AD-A085 652 KAPPA SYSTEMS INC COLORADO SPRINGS CO F/G 9/5

REPORT ON CONTRACT F05611-76-90203. VOLUME I.(U)

1976 G J GRIMES F05611-76-90203

UNCLASSIFIED NL

1 OF 1

AD A
085652

END
DATE
FILMED
7-80
DTIC

ADA085652

LEVEL III

2

FINAL REPORT

REPORT ON CONTRACT F05611-76-90203

Volume I of II

F05611 76 90203

11 1176 12 74

Prepared for:

Capt. D. K. Stevens, Chief
Inertial Guidance Research Division
The Frank J. Seiler Research Laboratory
USAF Academy, Colorado 80840

Prepared by:

Dr. C. J. Grimes
Kappa Systems, Inc.
1409 Potter Drive
Colorado Springs, Colorado 80909

391205

TABLE OF CONTENTS

Vol. I

Vol II

AD A085437

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to thank Mr. Bill Simmons, Captain David R. Stevens, Captain James M. Lind, Captain Perry Cole, TSgt Harold Haver, Airman Gary Lear and Airman Charles Coker for their significant contributions to their work.

I would also like to thank Capt. Roy Schmiesing and Mr. Michael N. Kunrod of DFEE and Dr. Gary Lamont of AFIT for their very helpful contributions in the early stages of this project.

SECTION 1

A BRIEF SUMMARY OF THE DIGITAL FILTER WORK

This report contains information on the digital filter work for isopad control performed in the past six month extension period of the contract. There is also some brief reports of the optical communication developments which have occurred during this period. These reports are included in this report only to make this a useful reference document, and no actual work was performed in this area during the past six month contract extention. All of the work was directed toward the microprocessor processor based digital filter control work for the isopad as directed by the Government in the SOW for the contract extension.

Of the four types of digital filters tested, the infinite impulse response filters showed the most promise for implementation in an isopad controller. The phase response in particular of these filters was very promising, and in simulation studies turned out to be even more promising than we expected at the onset of the project. However, because of delays through the processing network in real-time, the useful phase properties of the filters (particularly the phase lead over nearly two decades of frequency) could never be realized in real time, except in rare circumstances in which the digital filters were used in conjunction with some passive filters under unusual circumstances described in the following chapters.

I would recommend that this work be carried further by either using new modes of accessing the Digital Equipment Corporation LPS 11 system or by using a truly dedicated system. The software overhead of driving the Datel 256 was negligible, and if this same sort of arrangement could be realized with the analog to digital input device, whatever it might be and still preserve a very good stability of sample rate, which was achieved during these tests, perhaps the phase benefits of the digital filters could be realized in real-time.

Largely through the simulation plots obtained in this report, it was shown that the phase characteristics of the digital filters could be very useful in servo applications. These phase characteristics were very attractive. During the early stages of digital filter development, we were very skeptical of the phase results, partly because all of the available literature contained only information concerning the transfer function magnitude and specified very little

about the phase. We were also skeptical, because experts in the field had given us advice which consisted largely of warnings that in order to get lead we would have to implement infinite impulse responce filters, and that these filters would exhibit wildly fluctuation phase transfer functions and that because of this the filters would be useless or nearly so in real-time servo control of the isopad.

These fears were not founded. The filters showed very attractive phase characteristics. Even beyond this, even some inherently stable filters showed remarkably good phase characteristics, which would clearly not be obtainable in the analog implementations. This is apparently because of the nonanalyticity of the discrete time filters.

However, these filters were never proved to be effective in real-time isopad closed loop control tests because of difficulties in utilizing the lead. The effects of delays in the analog to digital process, the digital filtering, and the digital to analog process overshadowed the lead of the filters and gave lag.

Originally, we thought that we might benefit by changing the software to include such devices as optimal estimators to get the maximum amount of lead. This now appears not to be the case. If the filters described in this report can be implemented in hardware and software which minimized delay times through the processor, they show much promise.

The present pdp11/45-pdp11/03 combination still appears to be the best combination for microprocessor software development, and Captain Lind deserves much credit for this insight. This system shows much more promise than the National Semiconductor IMP-16 system originally procured for this purpose.

It might be wise to be aware of new sixteen bit microprocessors as they appear on the market. This is particularly true if sixteen bit microprocessors utilizing bipolar (I^2L, etc.) technology appear on the market and are supported with very good software development tools and are packaged on at least the board level with good I/O capability. But, for the present time, the LSI-11 microprocessor of the pdp11/03 appears to be a good if not the best choice.

One thing that was not tried since we could never get the filters to run fast enough anyway is the implementation of the filters in double precision arithmetic, this might help the phase characteristics, but since we could never

run fast enough even without this burden, this was never tried.

In conclusion, digital filters did show promise, but until large, not incremental improvements are made in the processing hardware and software, they cannot be utilized in real-time isopad seismic vibration control.

SECTION 2

PROJECT IDENTIFICATION

The main goal of this project was to explore ways in which digital filters could be used for isopad seismic vibration control. This goal was formulated after many analog circuits were built and tested to give the appropriate amount of phase lead in the proper frequency bands. None of the analog circuits had been totally successful in giving the required amount of lead. Unfortunately, in the literature of digital filters, there is very little, if any, useful information concerning the phase characteristics of digital filters. The primary goal was to explore the phase characteristics of digital filters relative to the isopad control problem.

The end goal of the isopad control problem is to execute various control loops, presumably one for each degree of freedom of the isopad, to control the isopad in real time using digital filters, eigher exclusively, or in conjunction with analog filters. It was originally concieved that the project would be carried out by developing software on the pdp11/45 and then down-line loading the program into the pdp11/03 computer which is based on the LSI-11 microprocessor. Thus, using microprocessor technology, it would be cost effective to use a controller per axis concept and the interaction or cross-coupling in the processors would be extremely well-defined by passing only the required parameters when needed between processors. This approach proved to be not feasible to put into practice because of the processing requirements of the filters. This will be explained in more detail in Section 14.

The original work plan for filter execution program development was to use FORTRAN IV plus programming on the pdp 11/45 in conjunction with the DEC optimized FORTRAN compiler (F4P). This would generate object code which will run on the pdp11/45 or the LSI-11 microprocessor in the pdp11/03 as efficiently as macro code written for either of these machines. The only real penalty of working like this is that the compilation times using the optimized compiler are a little longer, but this is an extremely mild penalty compared to writing the filter programs themselves in addition to the I/O drivers in assembly language. The approach was to do program development on the 11/45 and to test the programs using the LPS11 for data acquisition and the Datel 256 for the digital-to-analog converter function. After program development is complete for a single axis, it could be loaded by means of a forced down-line load from the 11/45 into the

11/03. After testing on the 11/03, it was thought to be desirable to "burn" the program in PROM's and install the PROM's in a separate LSI-11 processor board, thus freeing the 11/03 for further development work in the presence of the digitally controlled axis.

It was initially concieved that the programs to perform the digital filter coefficient synthesis tasks would be developed and run on the pdp11/45. This proved to be only partially ture. Of the two major types of digital filter coefficient synthesis programs, the finite impulse response and the infinite impulse response filters, only the finite impulse response programs would run satisfactorily on the 11/45. The infinite impulse response filter coefficient synthesis programs would never run properly on the pdp11/45. This program was more or less easily circumvented by using the Burroughs 6700.

The original plan called for implementing the following two types of filters:

1. Filters derived from the algorithms which were developed at AFIT under the direction of Dr. Gary Lamont.

2. Filters which were derived from the transfer function of the current analog control loops developed and tested by Mr. Bill Simmons at FJSRL.

The first approach was thought to be the most direct. However, a number of simplifying assumptions were made in the analysis which made the algorithms appear less attractive. The most serious of these assumptions used in the development of the algorithms was that the sensors in use are much better than they really are. This was not the fault of the AFIT personnel, since the algorithms they developed used the best information about the sensors which they had at the time.

The other approach would be to use a program which would synthesize all of the coefficients from a digital filter of a x the desired typeand degree from an experimentally obtained complex transfer function. The coefficients would be the synthesized to give the users choice of either flat as possible amplitude or flat phase or an optimal compromise of the two across the frequency spectrum of interest, probably 0.0001 Hz to 100 Hz.

The approaches which were actually tried were as follows:

1. Digital implementation as analog filters which were known to have a desirable transfer function.

2. Variable time delay filters.

3. Finite impulse response filters to give the desired phase response as indicated by the analog servo data.

4. Infinite impulse response filters to give the desired phase response as indicated by the analog servo data.

The first of these approaches was implemented by simply writing the difference equations for the various recursive filters and programming them. The second approach was done using a purely empirical technique. This technique consisted of putting in a software "time waster" loop and running this loop the maximum number of times for the experimentally determined lowest frequency likely to occur in a sample of a given time for a setting of the random noise generator.

The finite impulse response and infinite impulse response filters were desired using coefficient synthesis programs. Since no information exists regarding the phase of digital filters, we simply designed to the amplitude specifications that we thought we give the desired phase from the corresponding analog filter. This technique proved to be quite good most of the time. The validity of this technique depends, of course, on how closely the discrete time filter resembles a filter with a truly analytical transfer function. The finite impulse response filters were designed using a modification of a program from Theory and Applications of Digital Signal Processing by Lawrence Rabiner, Prentice Hall, 1975. The infinite impulse response filters were designed using the MAC/FIL software package purchased from Agbabian Associates which is discussed at some length later in this report.

SECTION 3

ANALYSIS FOR FILTER STABILITY AND FORMULATION OF DESIRED TRANSFER FUNCTION

A comprehensive review of digital filter and optimal control techniques has been completed relative to this problem. Information has been collected from a variety of textbooks, articles, and from the user's manuals of two major software packages obtained to synthesize digital filter coefficients for a particular type of digital filter. Other information has been obtained from Captain Roy Schmeising of DFEE, USAF Academy. Some of the most valuable information was obtained during my recent visit with Dr. Gary Lamont at the Air Force Institute of Technology (AFIT) at Wright-Patterson AFB, Ohio.

This study has resulted in a variety of techniques to try to improve the Isopad control. The trial solutions for this problem are as follows:

1. A simple second-order recursive low-pass filter with a corner frequency of about 25 Hz.

2. An FIR from the FIR filter synthesis program from "Theory and Application of Digital Signal Processing", by Rabiner and Gold.

3. An IIR filter designed with the use of the software package called MAC/FIL by Otnes of Agbabian Associates.

4. An optimal estimation technique used to increase the amount of phase lead in the region (1 to 25 Hz) in which it is needed.

Two basic systems approaches have been attempted using the computer system available at FJSRL. One used the Datel 256 system for the A/D input and the D/A output, and the other uses the LPS 11 (Laboratory Peripheral System) for these functions. Because of the strange nature of the interface card, the extremely poor documentation which accompanies the Datel 256 interface card for the DEC pdp 11/45, the Datel 256 system has not been successful. There are several problems, the most serious of which is that synchronous sampling must occur in a DMA mode. This would be a problem since it would not totally consume the operating system, an unfortunate occurrence in a multi-user situation. Even worse is the fact that the sampling must be done synchronously and the control output must be done asynchronously (to

minimize the time delay and thus the phase distortion between the sampling time and the time the output signal is sent to the shaker). This means that we must repeatedly initiate DMA transfers of one sample each. Even though this would probably work, it is clearly not desirable when others are trying to use the system. However, a program to accomplish this task has been written and could be installed if desired.

A FIR filter was implemented using the program from Rabiner and Gold. This program was supplied on punched cards by Dr. Gary Lamont of AFIT. It was installed on the pdp 11/45 system using the Burroughs 6700 system to write the file to 9-track magnetic tape, since the pdp 11/45 system does not have a card reader capability.

The FIR filter has several advantages. One is very predictable phase, which is linear. This makes it much easier to design with, since we are mostly designing in phase and the phase relationships of digital filters are largely unknown. To make matters worse, all of the software filter design packages use amplitude as the only design criteria. Thus, we must essentially use experience and empirical knowledge to guess the amplitude response of the desired corresponding phase response, in order to use the design packages. This is not too difficult if we are designing with a series of band-pass and band-reject filters, but it does present a problem when trying to synthesize a complex response across perhaps 5 decades of frequency.

The FIR filter is inherently stable, although this does not mean that the entire system will be stable. The major disadvantage is that it may not provide enough lead.

The implementation of the FIR filter has been done. After the coefficients were obtained for a 10th order filter using the Remez exchange algorithm, the execution program was simply edited into the same program previously written to execute the simple second-order recursive low-pass filter. A possibly more convenient technique could be used, namely that making the program that runs the timing and the LPS the main program and making the filter execution program a subroutine. However, this would increase the execution time. If an increase is to be made, it should be to increase the sampling rate or the precision of the multiplications, not for convenience.

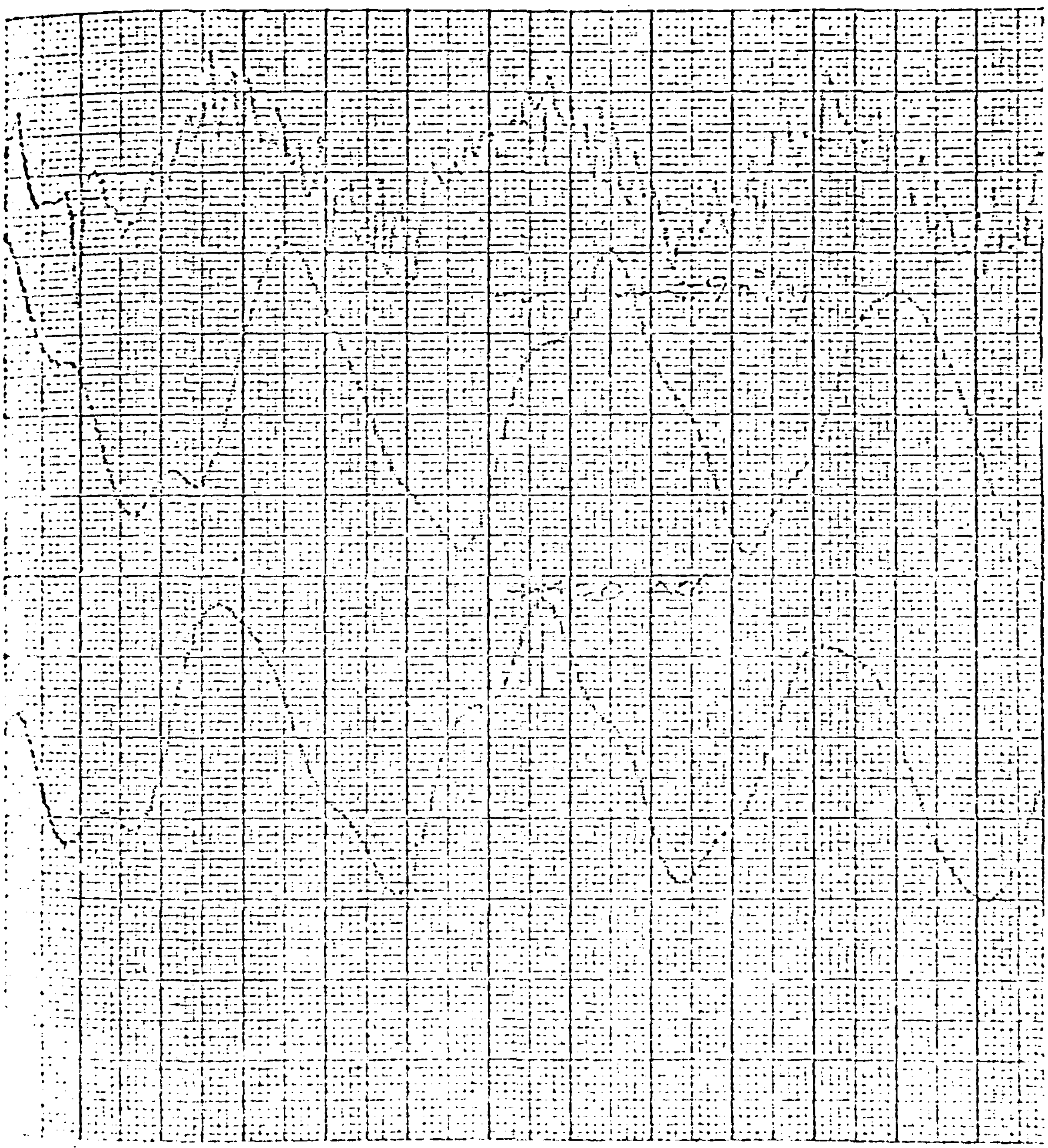
Single-precision arithmetic was used (with round-off) in the first trials. With the 16-bit machine, the errors would be a zero mean error of about 1/32,000 full-scale. Since we are looking for an improvement in the system of a factor of 100, this error is inconsequential.

The executions of the FIR filter can be expedited because it is a symmetrical filter. This means that we can first add the two appropriate sample values, multiply by the appropriate coefficient, and then add up the whole series to get the value of the output to convert to the analog signal to drive the shaker. Whether this approach is effective or not, the results will nevertheless be significant since little if anything is known about applying FIR filters to practical control problems.

The FIR filters, and perhaps even optimal estimator routines, may be needed to provide the desired phase. However, since the phase of these filters typically fluctuates wildly, it would be difficult if not impossible to predict in advance the total system response.

The MAC/FIL software package has been installed on the pdp 11/45 system, but it will not be useful until I redefine the logical unit numbers and modify it to run in an interactive mode. This will be a time consuming task (which is already partially completed) but will be worthwhile owing to the efficiency to true computer aided design (CAD) for actual filter design.

The isopad system transfer function is shown in figure 1. Also shown is the desired phase response for the first FIR filters. In the region from 0.01 Hz to 0.1 Hz the tiltmeter controls the servo. However, above 0.1 Hz the seismometer servo takes over and needs good gain, even at the resonant frequency, for good control. The two problem areas are the resonances around 1.5 Hz and 65 Hz. The obvious phase excursions are responsible for these problems. We attempted to remedy the phase response first rather than taking the more traditional approach of flattening the amplitude response.



[illegible]re 1 Demonstration that phase shift can be nollified in a post-processor [illegible] digital filter.

The work we have done which is summarized in this report shows how Mr. Bill Simmons and Dr. Gary Grimes have worked together to perfect a technique of synthesizing a digital filter of desired phase response by taking advantage of the analytic (in the complex variable sense given by the Cauchy-Riemann equations) between the magnitude and the phase portions of the transfer function. The technique basically consists of the following steps:

1. First the desired phase relationship is determined by using comparison techniques to the analog servo and by determining the areas in the frequency domain in which various degrees of lead and lag are needed to obtain maximum control of vibrations without making the system unstable or go into oscillation.

2. Next the magnitude portion of the transfer function is determined by using our experience with analog filters which would give the desired phase response.

3. Then the filter type is selected to give the desired transfer function magnitude response. This has been in recent tests either low-order recursive filters, finite impulse response (FIR) filters as synthesized by use of the Remez exchange algorithm, or variable time delay filters.

4. Then the filter is run and tested in real-time with the fast Fourier analyzer. Noise from the random noise generator is the input and the output is analyzed with respect to transfer function magnitude, transfer function phase, and coherence.

The remainder of this report will elaborate on how these four steps are used to empirically and experimentally arrive at the desired phase response for the iso-pad servo system.

SECTION 5

RECURSIVE FILTERS

The first type of filter to be tested is a simple first order recursive filter. It was of the lowpass type and have a cutoff frequency of about 25 Hz. This will, of course, limit the effectiveness of the servo to the lower frequencies. In order to achieve control out to 100 Hz we would presumably need a digital notch filter in the 55 to 65 Hz range to prevent oscillation of the system at the higher isopad resonance since this is what is presently required in the analog servo control system.

Thus, the next step would probably be a totally different (nonrecursive) type filter of the finite impulse response (FIR) type or the infinite impulse response (IIR) type. A series implementation of a lowpass filter with a cutoff frequency of about 100 Hz could be used in series with a band reject filter (with the rejection band very narrow with the center frequency about 60 Hz), but since we have programs which synthesize coefficients for easily realizable filters to do this same task, this will probably not be done.

A general form of the first order recursive filter is

$$y_i = \alpha y_{i-1} + g(x_i) \qquad (1)$$

where α is a constant parameter. In order for the filter to be stable the parameter α must be greater than -1 and smaller than 1.

If the transform G(f)X9(f) of $g(x_i)$ exists, then the complex transfer function is given by

$$H(f) = \frac{G(f)}{1 - \alpha \exp(-j2\pi\Delta t f)} \tag{2}$$

The lowpass filter has the form

$$y_i = \alpha y_{i-1} + (1-\alpha)x_i \tag{3}$$

where $g(x_i)$ is

$$g(x_i) = (1-\alpha)x_i \tag{4}$$

This results in a complex transfer function of

$$H(f) = \frac{1 - \alpha}{1 - \alpha \exp(-2j2\pi\Delta f)} \tag{5}$$

and an absolute value squared of the transfer function of

$$|H(f)|^2 = \frac{(1-\alpha)^2}{1 - 2\alpha \cos(2\pi f\Delta t) + \alpha^2} \tag{6}$$

The values of $|H(f)|^2$ for $f=0=1/2\Delta t$ are, respectively

$$|H(0)|^2 = 1 \tag{7}$$

$$\left|H\left(\frac{1}{2\Delta t}\right)\right|^2 = \left(\frac{1-\alpha}{1+\alpha}\right)^2 \tag{8}$$

The half power point of the lowpass filter is the frequency at which $H(f)^2$ has been reduced to one-half of the value it has at 0 Hz. If the half power point is at f_c Hz, then

$$\alpha = 2 - \cos 2\pi f_c \Delta t - \cos^2 2\pi f_c \Delta t - 4 \cos 2\pi f_c \Delta f + 3 \quad (9)$$

If we want to equate the simple first order recursive filter to a RC circuit in order to see the direct one-to-one relationship to the present analog servo, we only need to see that

$$\alpha = e^{-\Delta t/RC} \quad (10)$$

The analog servo is actually, of course, an active rather than a passive filter, but the RC time constant has exactly the same meaning and equation.

The measurements on which the design of the recursive lowpass filter was based is shown in figure 5-2. This figure shows both the phase transfer function of the isopad system without the control loop closed as well as the corresponding power of the log of the transfer function amplitude. The desired effect is to increase the closed loop lead between approximately 1 Hz where it begins to drop off and 50 Hz, which is the highest frequency of desired control since control above 50 Hz would necessitate the use of a band reject (notch) filter to avoid oscillation of the system at the higher isopad resonance. (The primary resonance is about 1 Hz.)

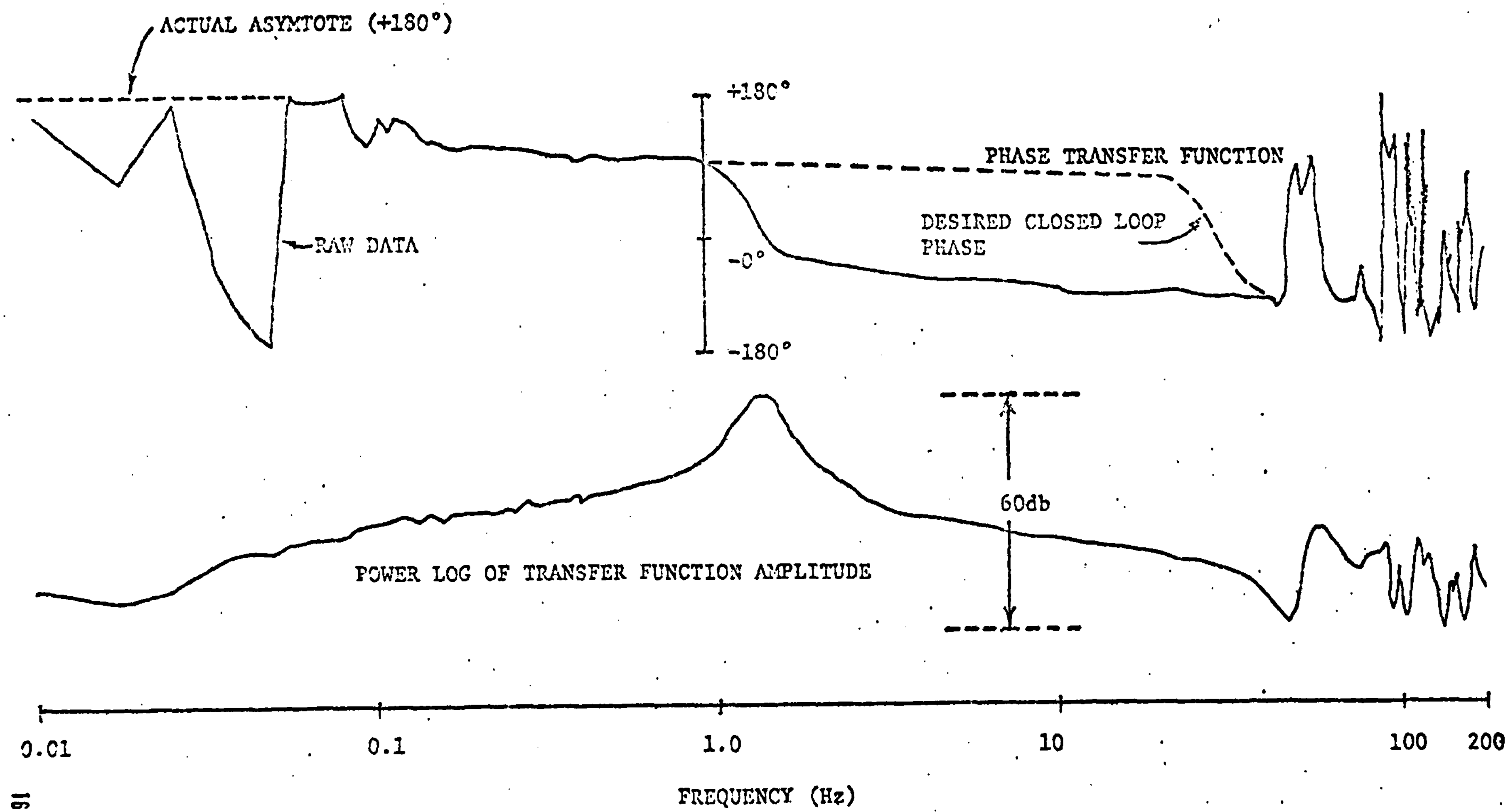


Figure 5-2. Transfer Function of FJSRL Iso-Pad (open loop) 6-6-77.

The original program (which is no longer operable since the DAC output of the LPS11 was transferred to the Aero Lab system) is shown in figure 3. It can be seen that this is a FORTRAN program which utilizes the LPS11 software and hardware to complete all sampling and control functions.

SECTION 6

HARDWARE SYSTEM CONFIGURATION

...ware system is shown in Figure 6-1. There are basically two ... inner real-time control loop and an outer post-processor loop ...alysis.

...entral processor the for inner real-time control loop was the ... The CPU for the post-processor analysis was the pdp 11/05 of ... Fast Fourier Analyzer system.

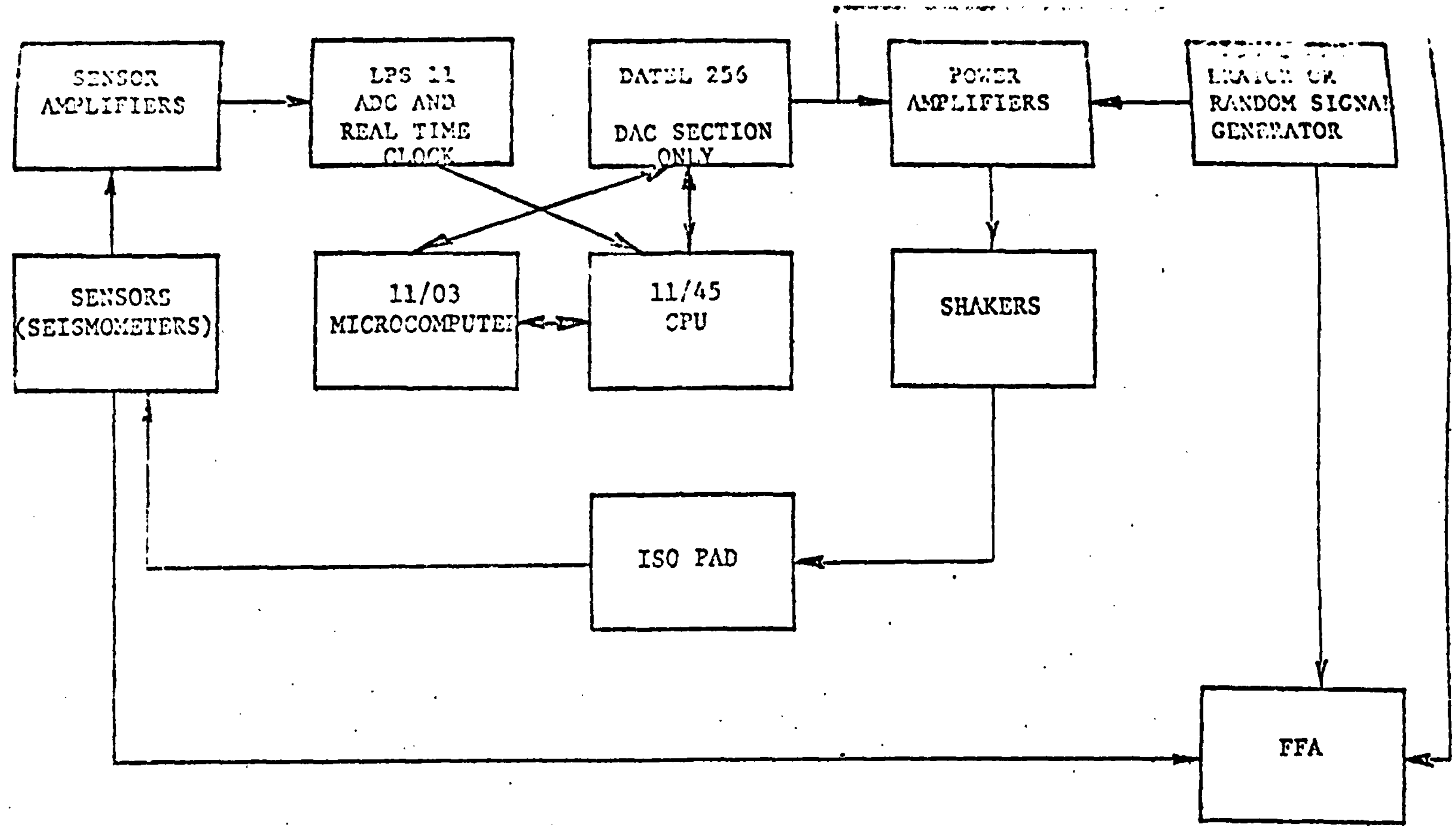


Figure 6-1. Wiring and Signal Flow Diagram for Digital Servo Design.

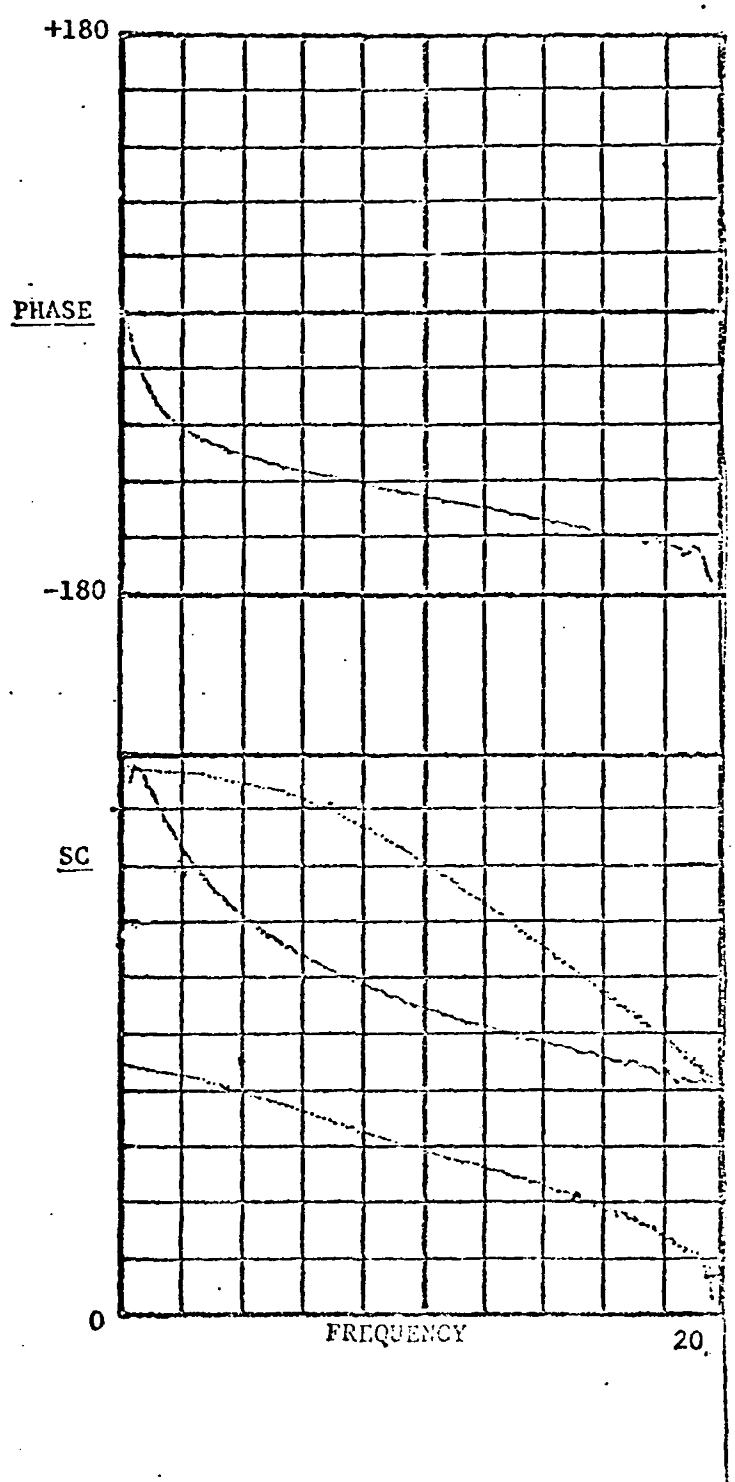
SECTION 7

VERIFICATION OF THE DIGITAL FILTER EXECUTION PROGRAM

The digital filter execution program was verified by using a version of it which simply executed a digital analog of a simple one-pole passive RC analog filter of the low pass type. The output of this filter was analyzed using the fast Fourier analyzer. The input was the output of the random noise generator with the output filtered below the Nyquist folding frequency to prevent aliasing. This was done simply to avoid putting an analog filter on the input.

The results are shown in figure7-1. The results show just exactly what might be expected and are nearly identical to what you would expect of a simple passive RC low pass filter. Since an arbitrary phase shift was not introduced into the digital filter the phase plot started at zero shift as one would expect.

The phase of this filter shows, of course, lag, and would be of little use for any purpose other than verifying the execution program.



re 7-1 Response of a simple digital filter which is similar to a one stage
ive RC low pass filter.

SECTION 8

THE VARIABLE TIME DELAY FILTERS

The next approach was to try a variety of variable time delay filters. Again, the main goal was to achieve a digital filter with servo lead. The idea here was to provide a variable time delay which was in some way inversely proportional to the frequency. This was done by allowing the computer to waste time according to a function which depended on the difference between the sample just taken and the sample taken previously. The larger the difference the smaller the time delay. So, for DC signals, the difference should be zero and the time delay at a maximum, and for some high frequency, the difference should be high and the delay should be equal to the minimum time delay permitted by the software execution time of the filter.

About a dozen time delay functions were tried with varying success. Several of them are shown in table 8-1. The constants in the time delay functions were adjusted at run time in an interactive fashion so that the maximum delay we were likely to get would be less than the time between samples. Otherwise the program would "bomb off" before the experimental data acquisition time of the Fourier analyzer was over and the data would be incomplete. This was done by selecting constants which were sure to cause a "bomb off" in a few seconds and scaling them down by 30% or so. Of course since the input signal was from a random noise generator, there was no guarantee that there would not be a particularly flat portion of the output signal which would still cause the program to bomb, in which case the Fourier analyzer would be started over again after the digital filter program was rescheduled.

The results are shown in figures 8-1 through 8-5. The phase of figure 8-2 is probably the most desirable, even though the slope is still negative. The data of figure 8-5 is a control. This shows that the software overhead causes a delay which prevents this program from accomplishing all of the desired goals. This is hoped to be avoided when the new operating system version is installed and the throughput is increased by a factor of 4.

Figure 8-6 shows graphically the relationships between the input and output signals for the variable time delay filters.

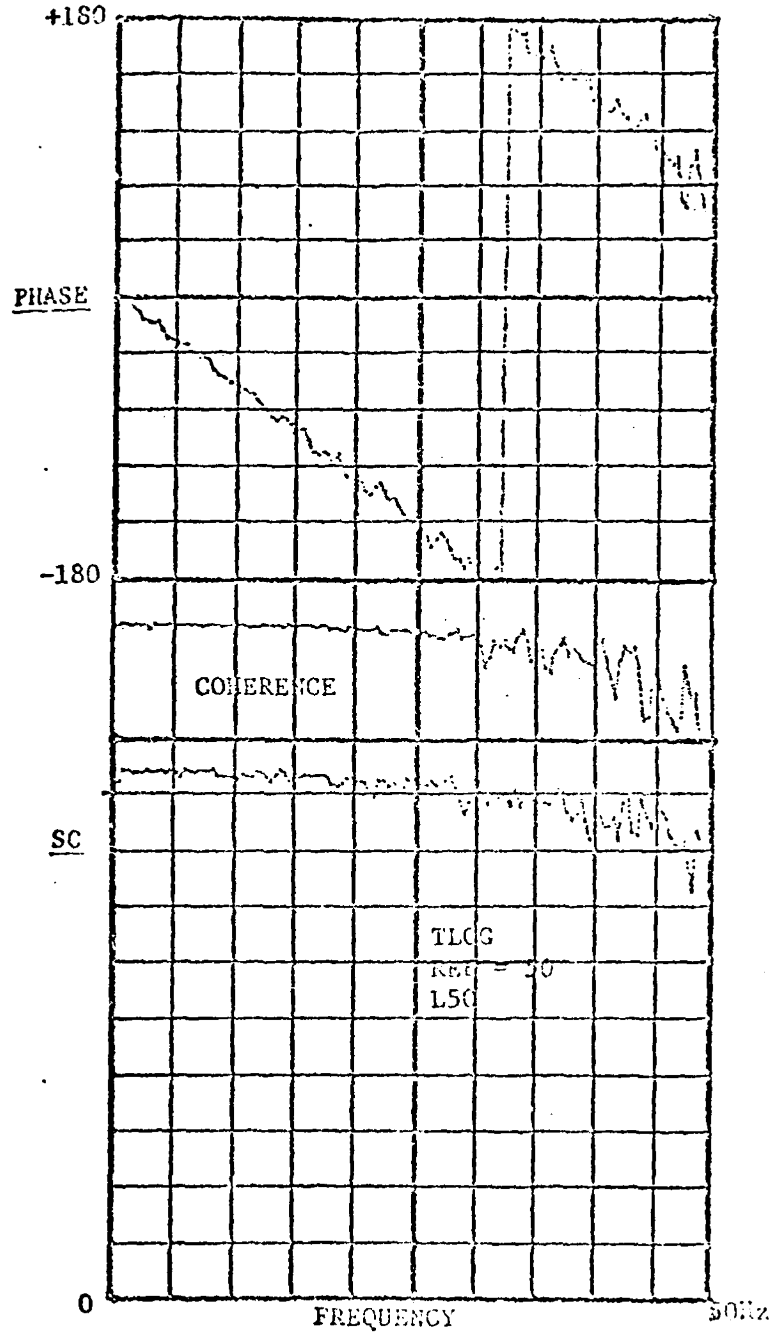
$$t_{delay} = t_{(delay)_{min}} + \frac{C_1}{C_2 + DIFF}$$

$$t_{delay} = t_{(delay)_{min}} + C_1 \times DIFF$$

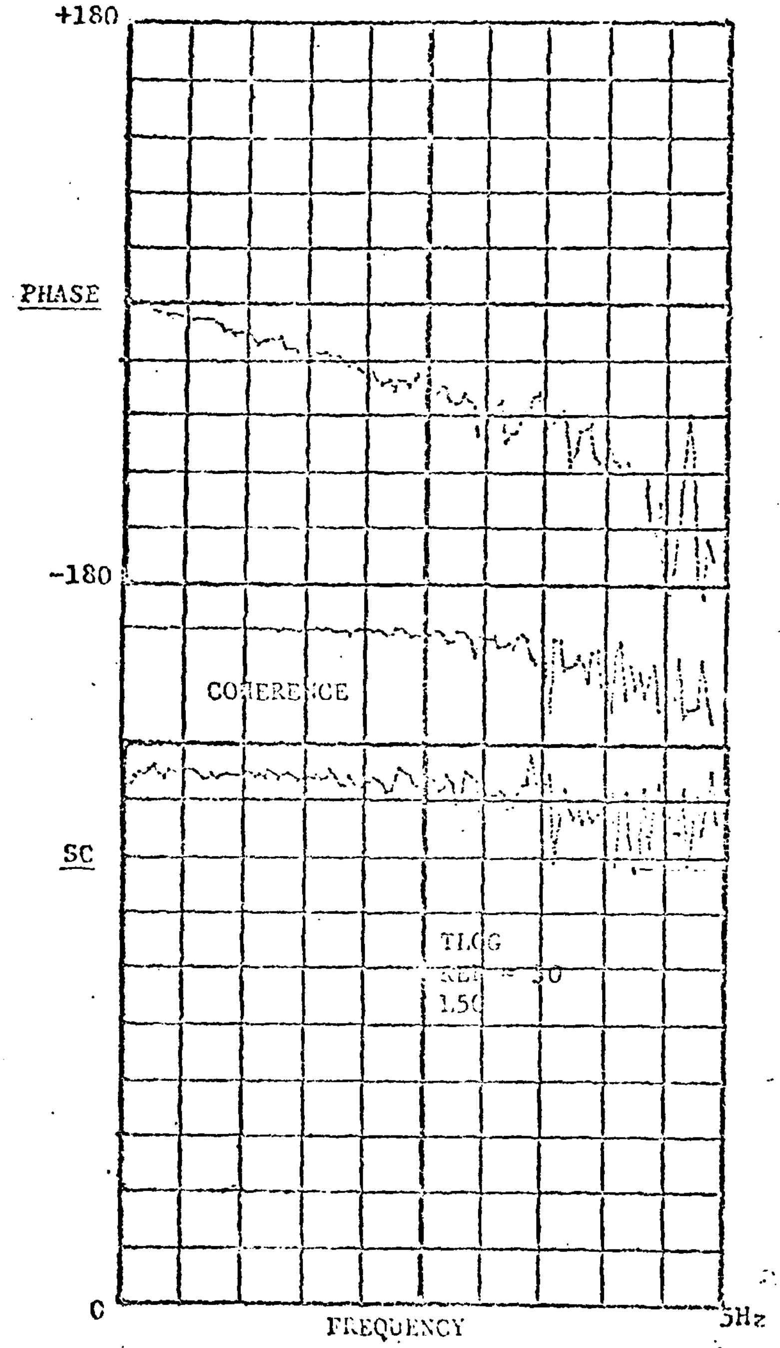
$$t_{delay} = t_{(delay)_{min}} + C_1 \times DIFF^2$$

$$t_{delay} = t_{(delay)_{min}} + C_1(C_2 - DIFF)^2$$

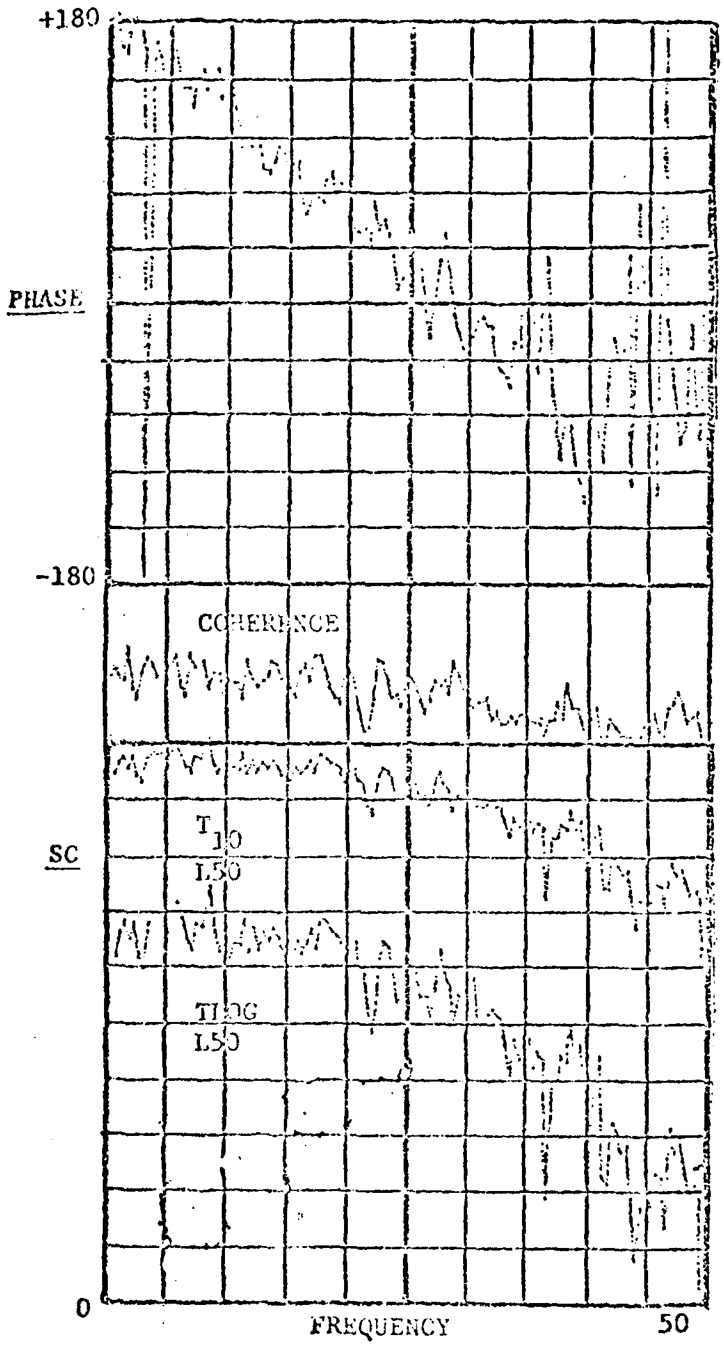
[illegible] 8-1 Samples of the equations for variable time delay digital filters.



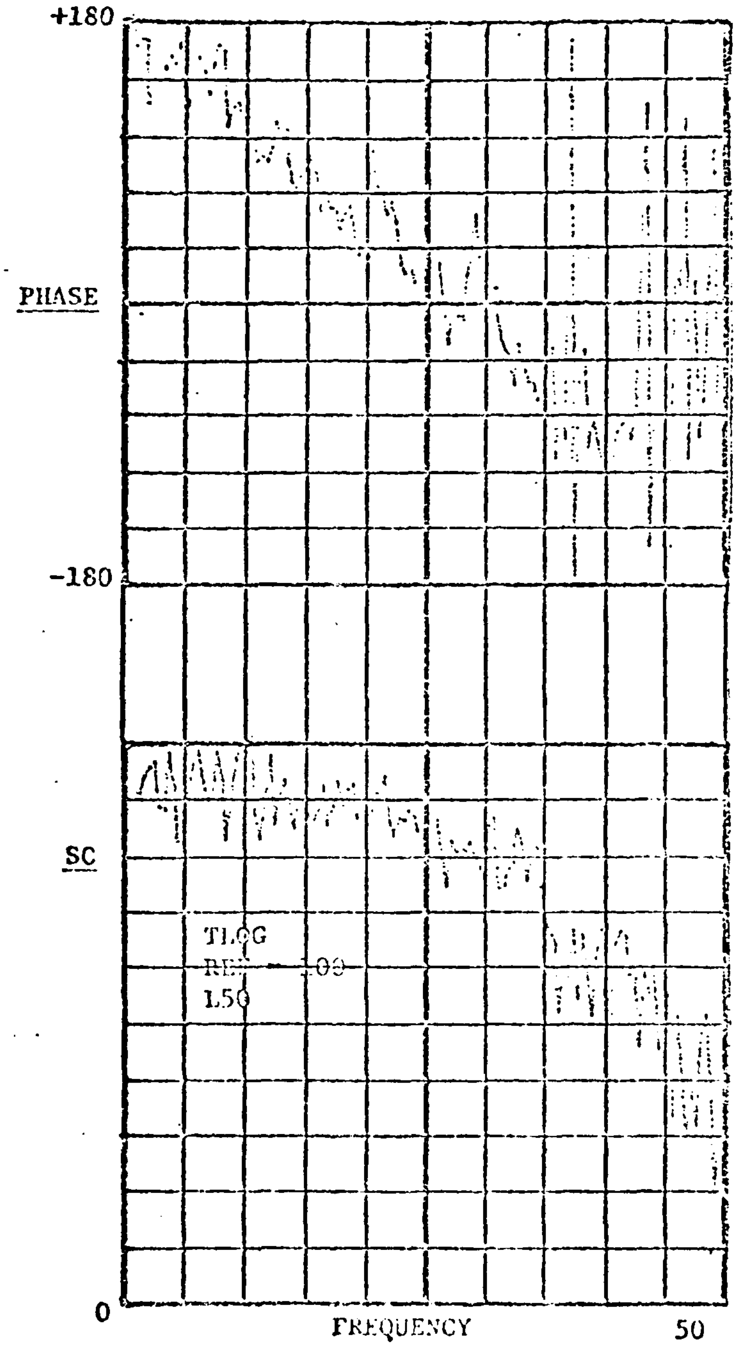
i Variable time delay digital filter with $t_{delay} = t_{(delay)_{min}} + C_j * DIFF$



2 Variable time delay digital filter with $t_{delay} = t_{(delay)_{min}} + \frac{C}{C_2 + DIFF}$



...iable time delay digital filter with $t_{delay} = t_{(delay)_{min}} + Cx(DIFF)^2$



...re 8-4 Variable time delay digital filter with $t_{delay} = t_{(delay)_{min}} + C_1[C_2 - DIFF]^2$

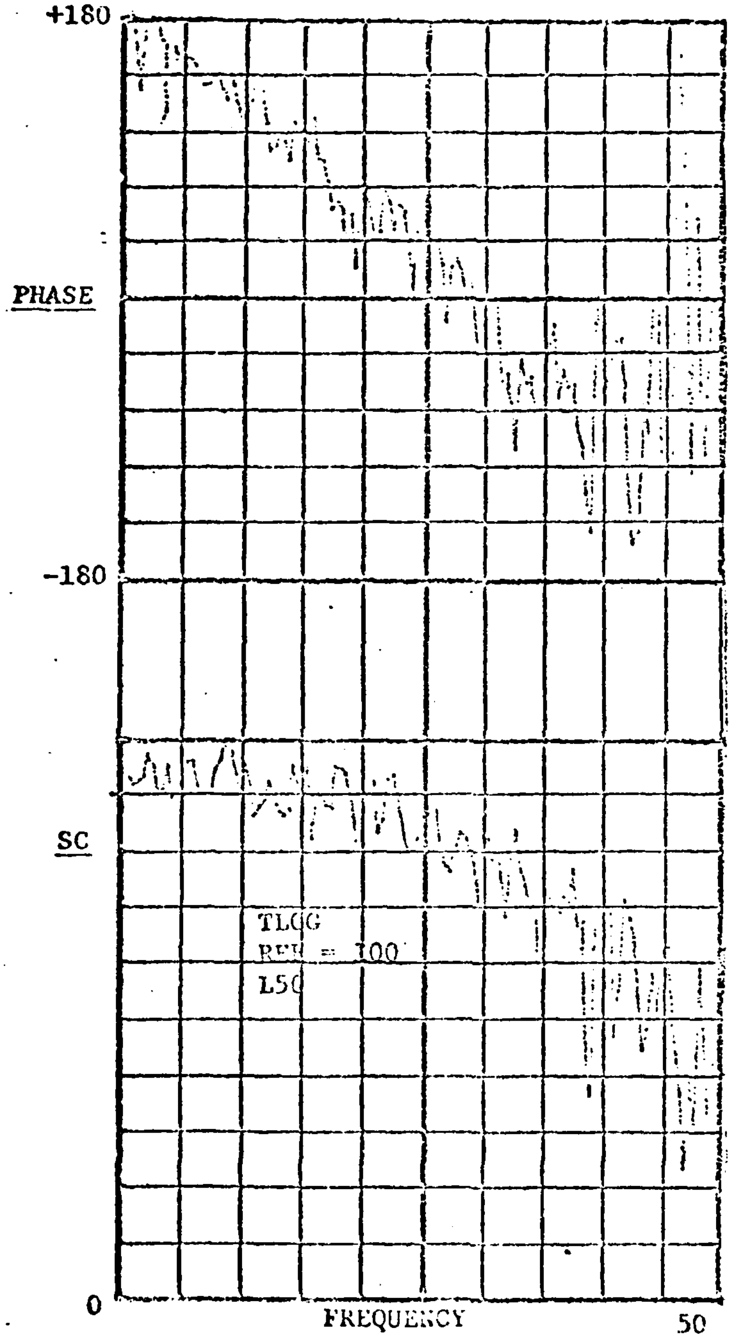


Figure 8-5 Control sample with fixed time delay.

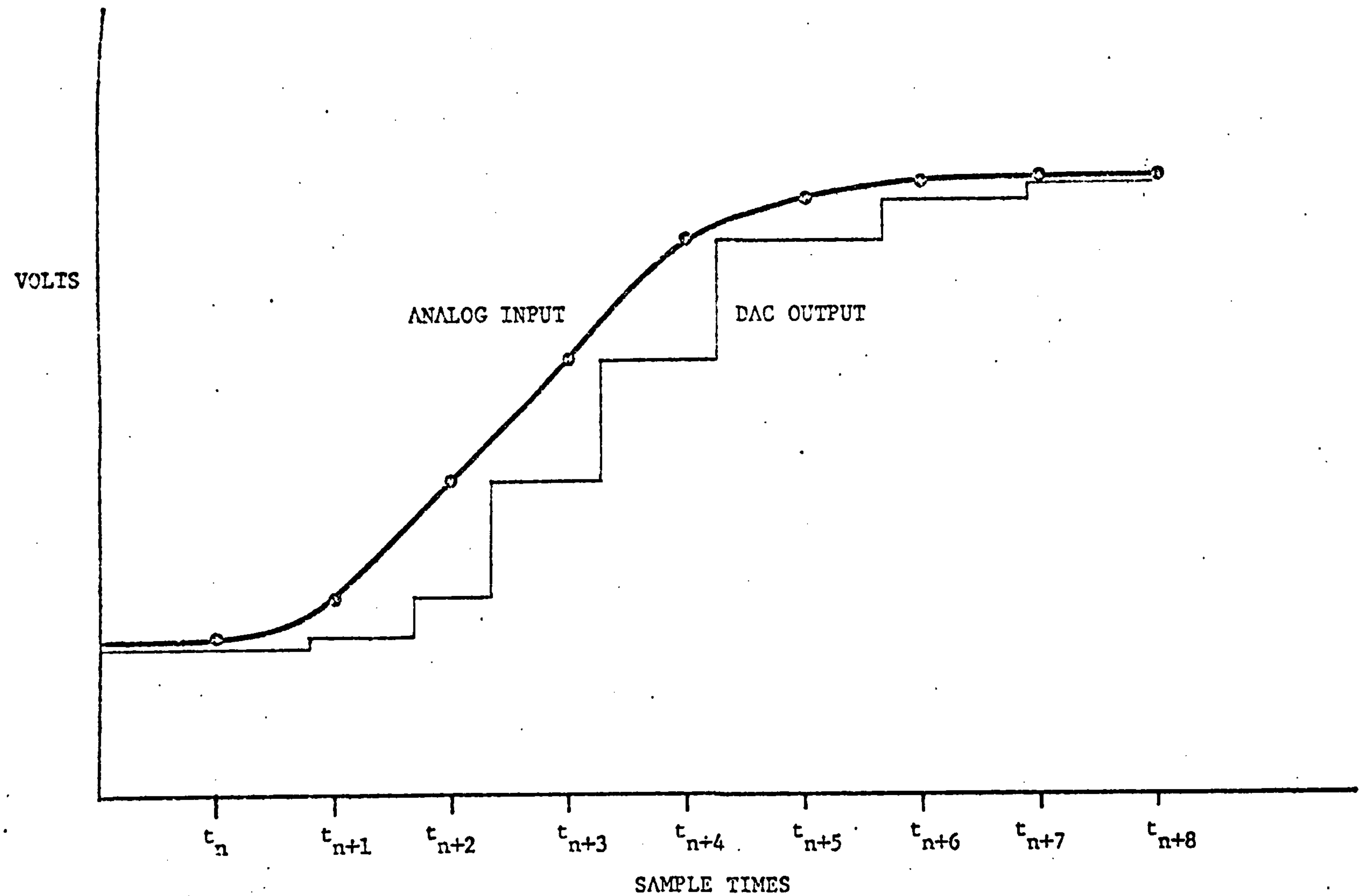


Figure 8-6 Variable time delay filter.

SECTION 9

THE INFINITE IMPULSE RESPONSE FILTERS

The FIR filter work has focused on applications of the FIR design program found in Rabiner and Gold's book "Theory and Application of Digital Signal Processing". This program is capable of designing a wide variety of optimal (minimax) FIR filters including lowpass, highpass, bandpass, and bandstop filters, as well as differentiators and Hilbert transformers.

A copy of this program has been installed on the pdp 11/45 and is ready to synthesize coefficients of first, a simple, lowpass, nonrecursive filter and then a filter having two passbands.

The implementation of the algorithms is quite simple one the coefficients have been found for the desired transfer function phase and magnitude. The implementation is simplified and shortened considerably in terms of execution times in the 11/45 and perhaps later on the LPS11 because it is a symmetric filter. This means that the coefficients are identical in pairs, the first equalling the last and so forth. This means that the execution is accomplished with an addition (or the two terms to be multiplied by the coefficient), a multiplication (of the coefficient times the previous results), and an addition to get the sum of all the terms. This means, for example, that a tenth-order filter requires only five multiplications on the machine, and this is important since the floating point multiplications take much more time than the additions, even with the newly installed floating point hardware.

If the H(x)'s are the coefficients and the Y(x)'s are the ...ted inputs, then the result to be output to the system for a ... order filter is

$$\text{Control output} = H(1)Y(1) + H(2)Y(2) + H(3)Y(3) + H(4)Y(4) + H(5)Y(5) + H(6)Y(6)$$

...ing the terms with identical coefficients we get

$$\begin{aligned}\text{Control output} = {} & H(1)[Y(1) + Y(6)] \\ & + H(2)[Y(2) + Y(5)] \\ & + H(3)[Y(3) + Y(4)]\end{aligned}$$

... writing it in series form we get

$$\text{Control output} = \sum_{I=1}^{N/2} H(I)[Y(I) + Y(N-I+1)]$$

... holds for any even number of coefficients, and the number of ...icients will always be even since it is a symmetric filter.

This means that the FORTRAN code to execute the filter, ...ing the synchronous sampling software and the asynchronous ...ly code which drives the Datel 256 system DACS, will look ...ng like the following.

```
      Y(1)  =  0.0
      Y(2)  =  0.0
            .
            .
            .
      Y(N)  =  0.0
      N  =  10
      H(1)  =  (coefficients)
      H(2)  =  (coefficients)
      H(N)  =  (coefficients)
 5    (Make new sample)
      DO 10 I  =  1, N-1
      Y(I+1)  =  Y(I)
10    CONTINUE
      Y(1)  =  new sample
      SUM  =  0.0
      DO 20  =  I = 1, N/2
      TERM = H(I)*[Y(I) + Y(N-I+1)]
      SUM  =  SUM + TERM
20    CONTINUE
      GO TO 5
      (output sum to DAC)
```

SECTION 10

THE FIR COEFFICIENT SYNTHESIS AND EXECUTION PROGRAMS

The program from Rabiner and Gold was modified so that it was compatible with the DEC PDP 11/45 processor and its optimized compiler (the F4P compiler). The enhanced version of FORTRAN IV was necessary because of the double precision functions and the required arccos function.

The program was changed so that it would operate in an interactive mode with the terminals of the PDP 11/45 system and put the output in hardcopy form to the Versatec printer. It also prints out many of the intermediate steps of the calculations at the terminal from which it is initiated for final coefficient verification. There are also a number of tracer comments which are printed out so that the exact order of execution of the program can be traced. If a hardcopy of this is desired, the program can be run from the Decwriter, although the amount of tracer comments and intermediate answers printed, delays the execution of the program considerably.

This program requires the user to specify the filter length, the type of filter (multiple passband, stopband, differentiator, Hilbert transformer), the number of bands, the grid density, the bandedges, the desired function for each band, and the weight function for each band.

Since the computer system time does not do anything and since the program does not take particularly long to execute anyway (several seconds), very high grid densities can be used for accuracy.

A quick check on the coefficients can be done on a calculator since the sum of them (taking into account how many times each of them is used) should be 1.00000000. The coefficients which are used twice (all the coefficients in an even length filter) should be multiplied by two. One coefficient (the "middle" one in an odd length filter) should only be used once in the sum.

Dr. Gary Lamont of AFIT, has run his version of the program and provided us with the coefficients of a nine length filter for a 25 Hz low-pass filter. The results are shown in table 1. These coefficients are also shown in the listing of program NINE.

The figures of this section show various runs for various parameters for FIR filters on the PDP 11/45 system. It can be seen that changes in the weighting and desired functions make the corresponding changes in the output parameters. The grid length was 16 unless stated otherwise in the captions.

At the end of this section is the modified version of the program. It can be seen that it is interactive. The interactive version is named ASKFIR. Another interactive version is simply FIR, but this version does not prompt the user with comments for the input parameters. In all other respects these two versions are identical.

```
                    FINITE IMPULSE RESPONSE (FIR)
                    LINEAR PHASE DIGITAL FILTER DESIGN
                    REMEZ EXCHANGE ALGORITHM

                    BANDPASS FILTER

          FILTER LENGTH =  32

          ***** IMPULSE RESPONSE *****
               H(  1) = -0.59143951E-01 = H(  32)
               H(  2) =  0.92645919E+00 = H(  31)
               H(  3) = -0.73567076E+01 = H(  30)
               H(  4) =  0.39310806E+02 = H(  29)
               H(  5) = -0.15833011E+03 = H(  28)
               H(  6) =  0.51045930E+03 = H(  27)
               H(  7) = -0.13561403E+04 = H(  26)
               H(  8) =  0.31053657E+04 = H(  25)
               H(  9) = -0.60023179E+04 = H(  24)
               H( 10) =  0.10393407E+05 = H(  23)
               H( 11) = -0.15507367E+05 = H(  22)
               H( 12) =  0.20153451E+05 = H(  21)
               H( 13) = -0.22564871E+05 = H(  20)
               H( 14) =  0.21113258E+05 = H(  19)
               H( 15) = -0.15143297E+05 = H(  18)
               H( 16) =  0.55170391E+04 = H(  17)

                          BAND  1           BAND
LOWER BAND EDGE      0.000100000
UPPER BAND EDGE      0.250000000
DESIRED VALUE        1.000000000
WEIGHTING            1.000000000
DEVIATION            0.000000028
DEVIATION IN DB -150.904403687

EXTREMAL FREQUENCIES
  0.0020531   0.0137719   0.0313500   0.0450219   0.0626938
  0.0752953   0.0918969   0.1055453   0.1260766   0.1397719
  0.1541625   0.1687641   0.1963891   0.2129985   0.2286155
  0.2461938   0.2481469
```

Table 10-1 These are the coefficients for a length 32 filter with cutoff frequencies 0.2 Hz and 25 Hz.

```
FINITE IMPULSE RESPONSE (FIR)
LINEAR PHASE DIGITAL FILTER DESIGN
REMEZ EXCHANGE ALGORITHM

BANDPASS FILTER.

FILTER LENGTH =  24

***** IMPULSE RESPONSE *****
        H(  1) = -0.11551735E-02 = H(  24)
        H(  2) =  0.12359285E-01 = H(  23)
        H(  3) = -0.72669894E-01 = H(  22)
        H(  4) =  0.27635461E+00 = H(  21)
        H(  5) = -0.78837645E+00 = H(  20)
        H(  6) =  0.17829072E+01 = H(  19)
        H(  7) = -0.32935483E+01 = H(  18)
        H(  8) =  0.50383058E+01 = H(  17)
        H(  9) = -0.63789263E+01 = H(  16)
        H( 10) =  0.65435295E+01 = H(  15)
        H( 11) = -0.50767002E+01 = H(  14)
        H( 12) =  0.24573517E+01 = H(  13)

                     BAND  1          BAND
LOWER BAND EDGE    0.000000000
UPPER BAND EDGE    0.250000000
DESIRED VALUE      1.000000000
WEIGHTING          1.000000000
DEVIATION          0.000000022
DEVIATION IN DB -153.065277100

EXTREMAL FREQUENCIES
  0.0000000   0.0156250   0.0364583   0.0625000   0.0781250
  0.1067708   0.1197916   0.1484375   0.1770834   0.1979167
  0.2187501   0.2447918   0.2500000
```

Table 10-2 Since the lower cutoff is 0.0 Hz, this is really a 25 Hz low-pass filter. Notice that there are some coefficients larger than one, but the sum is still just one. This is necessary for stability.

$$H(1) = -0.54497945 = H(9)$$

$$H(2) = 0.2021985 = H(8)$$

$$H(3) = -0.59043782 = H(7)$$

$$H(4) = 0.32028612 = H(6)$$

$$H(5) = 0.55120021$$

Table 10-3 Coefficients for a length 9 FIR filter from Dr. Gary Lamont of AFIT.

```
                FINITE IMPULSE RESPONSE (FIR)
                LINEAR PHASE DIGITAL FILTER DESIGN
                REMEZ EXCHANGE ALGORITHM

                BANDPASS FILTER

          FILTER LENGTH =  10

          ***** IMPULSE RESPONSE *****
               H(  1) =  0.22404280E-02 = H( 10)
               H(  2) = -0.14221852E-01 = H(  9)
               H(  3) =  0.52395659E-01 = H(  8)
               H(  4) = -0.15598708E+00 = H(  7)
               H(  5) =  0.61552787E+00 = H(  6)

                    BAND  1          BAND
LOWER BAND EDGE    0.000100000
UPPER BAND EDGE .  0.250000000
DESIRED VALUE      1.000000000
WEIGHTING          1.000000000
DEVIATION          0.000060355
DEVIATION IN DB  -83.307098309

EXTREMAL FREQUENCIES
  0.0001000   0.0526900   0.1313500   0.1876003   0.2313499
  0.2500000
```

Table 10-4 These are the coefficients for a length 10 FIR filter which passes signals in the band 0.0001 to 0.25. For our sample rate of 100 Hz, this is a 0.01 Hz to 25 Hz bandpass filter. The grid length used here was 16.

```
                    FINITE IMPULSE RESPONSE (FIR)
                    LINEAR PHASE DIGITAL FILTER DESIGN
                    REMEZ EXCHANGE ALGORITHM

                    BANDPASS FILTER

          FILTER LENGTH =  10

          ***** IMPULSE RESPONSE *****
               H(  1) =  0.21503528E-02 = H( 10)
               H(  2) = -0.13926538E-01 = H(  9)
               H(  3) =  0.51863350E-01 = H(  8)
               H(  4) = -0.15535874E+00 = H(  7)
               H(  5) =  0.61528701E+00 = H(  6)

                     BAND  1            BAND
LOWER BAND EDGE    0.000100000
UPPER BAND EDGE    0.250000000
DESIRED VALUE      1.000000000
WEIGHTING          1.000000000
DEVIATION          0.000063005
DEVIATION IN DB  -84.012458801

EXTREMAL FREQUENCIES
  0.0001000   0.0719750   0.1391625   0.1969752   0.2358379
  0.2469754
```

Table 10-5 This filter is identical to the filter of Table 2, except the grid length is 64. Notice the coefficients are slightly different.

```
                    FINITE IMPULSE RESPONSE (FIR)
                    LINEAR PHASE DIGITAL FILTER DESIGN
                    REMEZ EXCHANGE ALGORITHM

                    BANDPASS FILTER

          FILTER LENGTH =  10

          ***** IMPULSE RESPONSE *****
               H( 1) =  0.20724945E-02 = H( 10)
               H( 2) = -0.13615716E-01 = H(  9)
               H( 3) =  0.51299512E-01 = H(  8)
               H( 4) = -0.15476017E+00 = H(  7)
               H( 5) =  0.61503077E+00 = H(  6)

                          BAND  1         BAND
LOWER BAND EDGE      0.000100000
UPPER BAND EDGE      0.250000000
DESIRED VALUE        1.000000000
WEIGHTING            1.000000000
DEVIATION            0.000053613
DEVIATION IN DB    -85.414619446

EXTREMAL FREQUENCIES
  0.0001000   0.0719750   0.1375999   0.1907249   0.2344749
  0.2438439
```

Table 10-6 This is another 0.01 Hz to 25 Hz bandpass filter, but with a grid length of 32.

```
                    FINITE IMPULSE RESPONSE (FIR)
                    LINEAR PHASE DIGITAL FILTER DESIGN
                    REMEZ EXCHANGE ALGORITHM

                    BANDPASS FILTER

              FILTER LENGTH =  24

              ***** IMPULSE RESPONSE *****
                   H(  1) =  0.18263933E-03 = H(  24)
                   H(  2) = -0.19541905E-02 = H(  23)
                   H(  3) =  0.10693687E-01 = H(  22)
                   H(  4) = -0.39573073E-01 = H(  21)
                   H(  5) =  0.11037862E+00 = H(  20)
                   H(  6) = -0.24499482E+00 = H(  19)
                   H(  7) =  0.44530298E+00 = H(  18)
                   H(  8) = -0.67153382E+00 = H(  17)
                   H(  9) =  0.83729565E+00 = H(  16)
                   H( 10) = -0.83935499E+00 = H(  15)
                   H( 11) =  0.60951639E+00 = H(  14)
                   H( 12) = -0.11603920E+00 = H(  13)

                        BAND  1          BAND
 LOWER BAND EDGE    0.001000000
 UPPER BAND EDGE    0.250000000
 DESIRED VALUE      0.200000003
 WEIGHTING          0.800000012
 DEVIATION          0.000000004
 DEVIATION IN DB -167.807632446

 EXTREMAL FREQUENCIES
   0.0010000   0.0068125   0.0340542   0.0538833   0.0817292
   0.1103750   0.1260000   0.1624504   0.1885801   0.2119376
   0.2327710   0.2485835   0.2500000
```

Table 10-7 This is a length 24 bandpass filter with cutoff frequencies 0.1 Hz and 25 Hz. The grid length was 16.

Both major types of filter execution programs were installed on the PDP 11/45 system and their operation was verified at useable sample rates using the light emitting diode digital displays on the laboratory peripheral system (LPS11). All of the programs would run at sample rates over 100 Hz. This was determined to be adequate since it is over three times the frequency of interest of the first filters we will try which cut off below 30 Hz (probably at 25 Hz).

The stability of all of these filters was also verified by allowing them to run for extended periods at various setting of the potentiometers on the LPS11.

There are three execution programs which currently run on the 11/45 system. Two of them execute finite impluse response (FIR) filters. One of these is for filters of even length and one is for filters of odd length. The third filter execution program is for a first order recursive low-pass filter corresponding to a one-stage passive analog RC filter.

The even length filter execution program has been tested for length ten filters and the odd length filter program has been tested for nine length filters. Both of these would operate with a sample rate of 100 Hz.

The recursive filter execution program was designed to be non-interactive since it only requires that the sample rate and the low-pass corner frequency of -3 db cutoff frequency be entered at execution time. The FIR filter execution programs do not run in an interactive mode since too many filter coefficients need to be changed for various parameters and this would make the procedure very cumbersome.

The DEC optimized compiler was used to compile all of the filter execution programs.

The following sequence was used to compile, build the tasks, and begin the real-time filtering routine:

```
F4P EASY=EASY
TKB EASY/PR: 4 = EASY, IDAC, [1,1] F4POTS/LB
RUN EASY/PRI= 150
```

Different programs were written for FIR filters of odd and even length ... minimize the run time and thus increase the sample rate since is is somewhat ...ginal.

The somewhat marginal sample rate is caused by the software overhead used ... the 11/45 system to operate the laboratory peripheral system. It is not ...ited by the efficiency of the compiler or the speed of the hardware floating ...int processor.

Two major changes in the system configuration were under serious con-...ideration. These were:

1. Transferring the processor task to the PDP 11/03 (containing the LSI-11 microprocessor and 12 K of memory).

2. Letting the Datel 256 system handle all of the analog to digital interfacing.

The first of these tasks required that special drivers be written to ...trol the laboratory peripheral system if it is still in use. The software ...rhead required by the laboratory peripheral system, if it is used in the ...sent FORTRAN callable configuration, would simply overwhelm the PDP 11/03.

Setting the sample rate with software routines will simply not be accurate ...gh since even the slightest amount of jitter in the sample rate will intro-...e spurious frequencies into the filter. Even though an analog anti-aliasing ...ter will have to be used in front of the digital filter, it cannot eliminate ... spurious frequencies introduced. A digital anti-aliasing filter is by ...inition impossible. Even if the sample rate were absolutely uniform.

The FIR execution programs are perhaps the most promising filters to be ...ed in the near term since they are linear filters and the linear phase not ... makes them inherently most stable than filters with wildly varying phase ...s infinite impulse response-IIR-filters) but also makes phase the ...y design criteria. This is important since the system stability can be ...ined far more easily if phase rather than amplitude is used at the ...l stages of filter design.

This technique allowed us to make a quick determination of the worth of the digital filters since the main problem with the analog filters over the past two years has been not being able to design enough lead into the filters without making the system unstable. With the FIR linear phase technique we were able to make the system stable by making the lead increase very gradually over frequency and then drop off to lag very quickly by changing the weighting functions in the various bands.

The desired result of this in simple terms is shown in figure 10-1. It can be seen that the phase increases toward the lead side very slowly, and then drops off very quickly at a point in the vicinity of an isopad resonance (probably a flexing resonance). In order to maintain the proper correspondence between the phase and the amplitude function the amplitude will probably resemble the amplitude portion of figure 10-1. This is because we assume that the transfer function is an analytic function (in the complex variable sense) and that for all analytic functions, once the phase is determined, the magnitude portion of the transfer function is automatically determined. It is also true that once the magnitude portion is determined, the phase portion is automatically determined, and this is the "backdoor" approach to filter design that has been used in the past.

We could not verify this analyticity for discrete-time (digital or "sampled") filters. After discussing this problem with a number of experts in the field of digital filtering, it is clear that there is a great deal of confusion over the analyticity of digital filters.

The most probable result will be that the analyticity is maintained much as in the analog case except for the fact that the phase can be arbitrarily shifted. This means that the phase can be tailored as needed at any arbitrary frequency. However, the catch appears to be that the entire frequency response transfer function must be shifted along the frequency as a whole and that this may introduce a filter instability, and thus a system oscillation, at an isopad resonance.

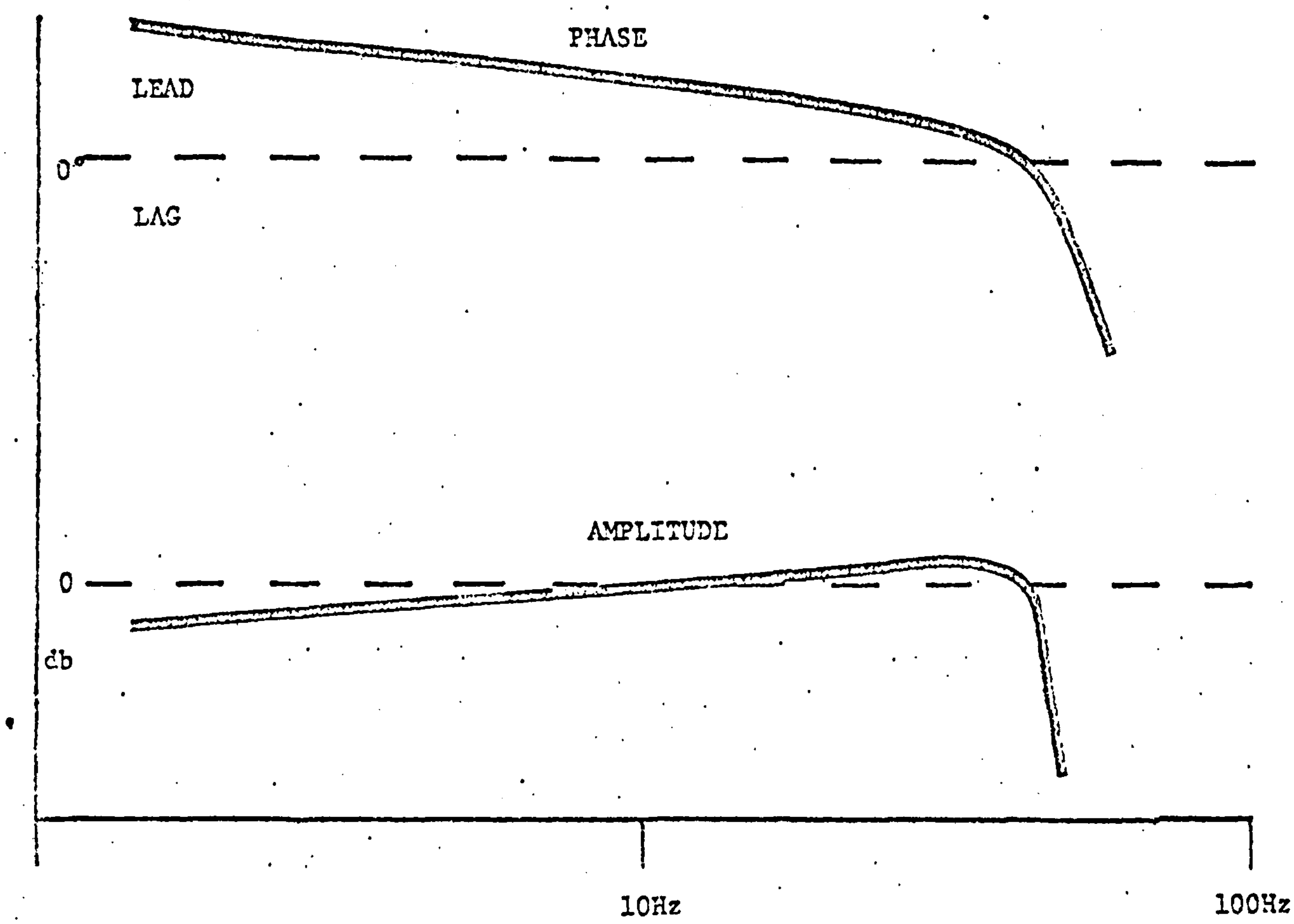


Figure10-1 Designed response of modified bandpass FIR filter.

SECTION 11

THE FIR FILTER RESULTS

The first FIR filter results were obtained on a nine length filter whose ...icients were generated by Dr. Gary Lamont at AFIT, Wright-Patterson AFB, These coefficients are shown in table 11-1.

The results of Fourier analysis are of this filter are shown in figure 3. ...ilter produced generally the results that were expected, that is a gradual ...off at a frequency 0.25 times the sample frequency. The filter was ...ally designed to be a 25 Hz low pass filter, but in order to reduce the ...ts of the fixed delay of the filter we ran it at a 10 Hz sample rate ...r than a 100 Hz sample rate as originally planned. This produced the ... 2.5 Hz cutoff.

One of the unexpected results was the tremendously rough nature of the ...fer function. There are two possible explanations of this. One is that ...sample rate of the Fourier analyzer was roughly twice that of the digital ...er. Thus for some outputs of the digital filter, the number of samples ... be two, while in other cases it could have been one or three. Since the ...le rates were very nearly even multiples, there could have even been ...plained difference frequency effects, which could have caused the rough ...e of the plots. The other possible explanation offered by Dr. Gary Lamont ... that a ninth order filter is usually not expected to be very smooth. ...ver, by comparing these plots with those of typical pass-band and stop-band ...le in various books on digital filtering, this explanation seems unlikely, ...cially since Dr. Lamont offered this suggestion without the benefit of ...ing the plots.

The second FIR filter to be tested was a 32 stage differentiator. It was ...d that this filter would exhibit lead. By this we mean lead in the servo ...ext, that is the plot of phase versus frequency should have a positive ... In the usual context lead merely means that the phase relationship is ...tive. This filter exhibited a positive phase relationship over a portion ...its spectrum, but it was disappointing to the extent that nowhere did it ...bit lead in the servo sense.

The coefficients for this filter are shown in table 11 2.

```
SET H COEFFICIENTS
H(1)=-.54[illegible]
H(2)=-.2[illegible]
H(3)=-.5[illegible]
H(4)=.32[illegible]
H(5)=.5512[illegible]
```

Table 11-1 Coefficients for a length 9 FIR low pass filter.

```
SET H COEFFICIENTS
H(1)=-0.62713051
H(2)= 0.85633453E-03
H(3)=-0.42418549E-03
H(4)= 0.39981510E-03
H(5)=-0.43437273E-03

H(6)= 0.49939450E-03
H(7)=-0.59634961E-03
H(8)= 0.73277031E-03
H(9)=-0.93032501E-03
H(10)= 0.1227042E-02
H(11)=-0.17012820E-02
H(12)= 0.25272341E-02
H(13)=-0.41601150E-02
H(14)= 0.81294555E-02
H(15)=-0.22339037E-01
H(16)= 0.20265539E 00
```

Table 11-1 Coefficients for a length 32 FIR differentiator.

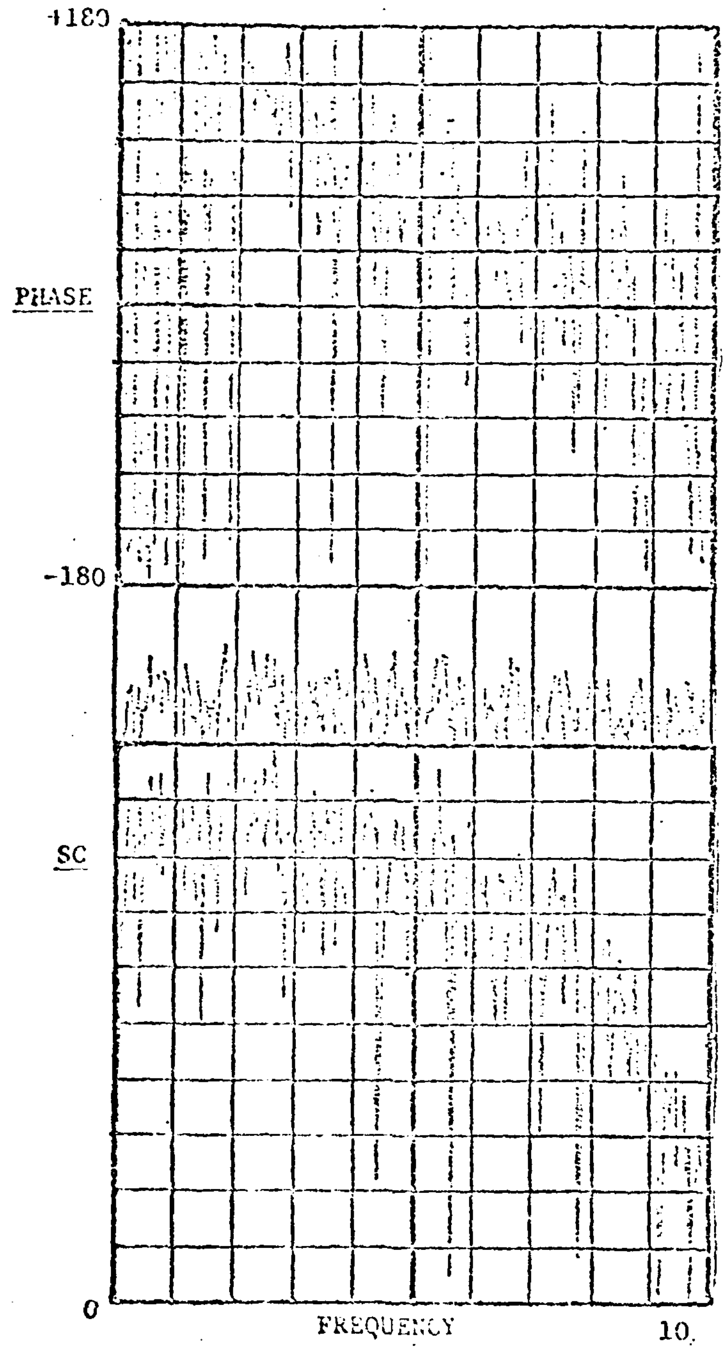


Figure 11-2 A 9 stage FIR low pass filter of 10hz cutoff.

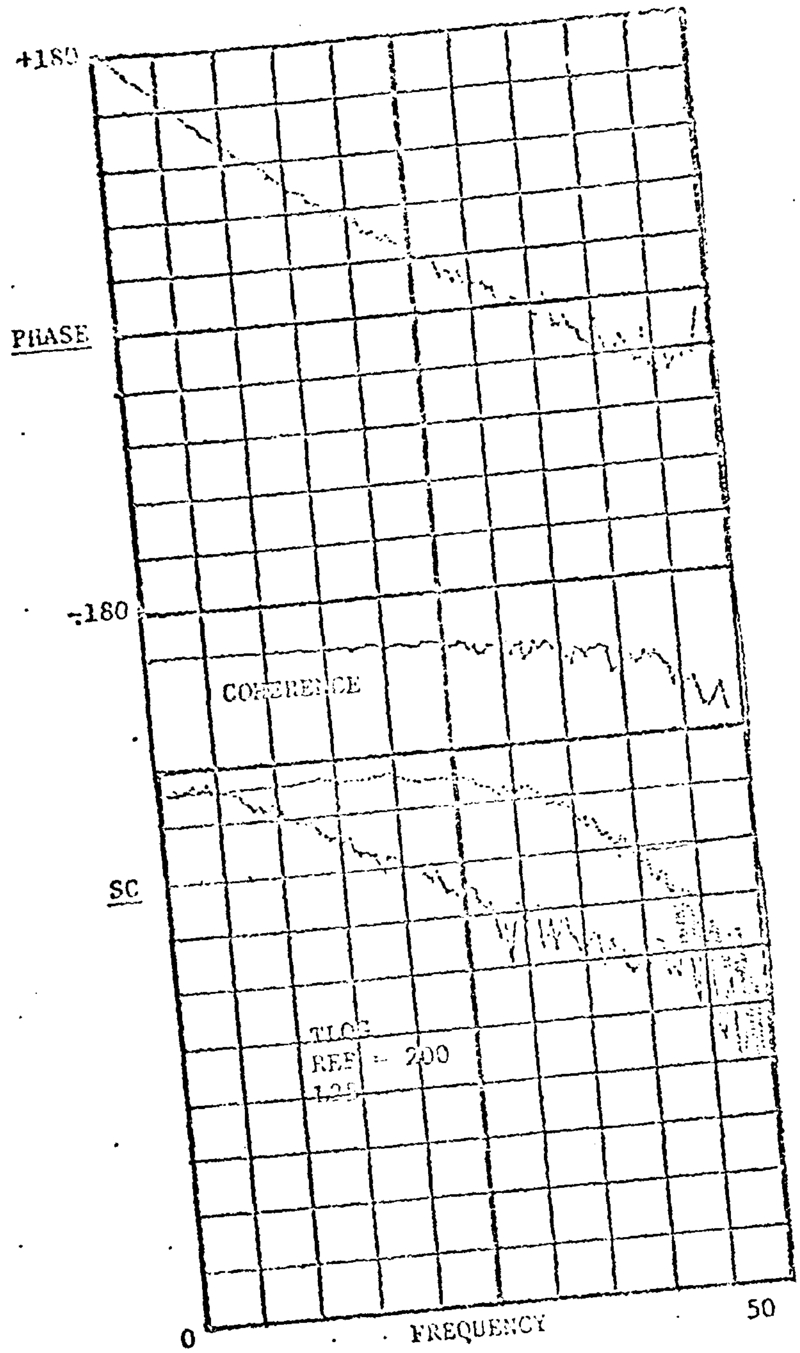
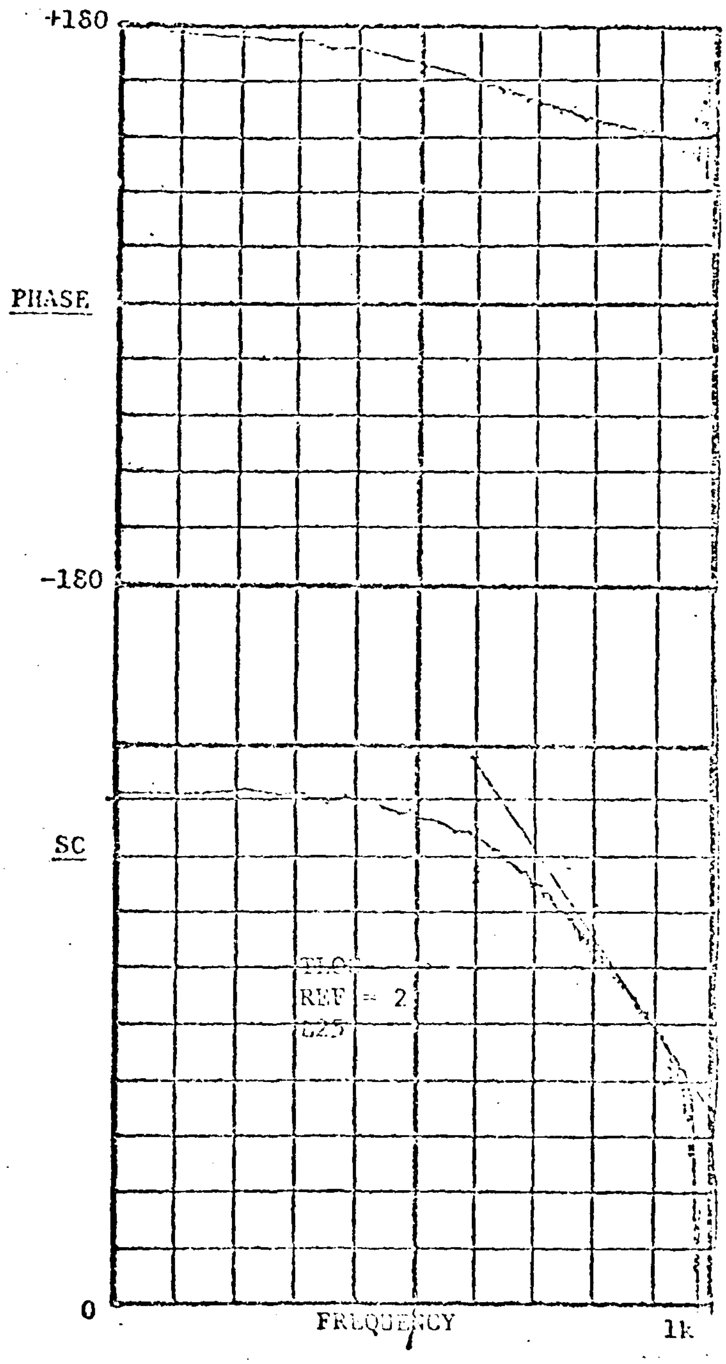


Figure 11-3 Digital FIR filter 32nd order differentiator.



...ure 11-4 Response of the digital filter of Figure 4 with a smoothing fil...

A perhaps somewhat unexpected result was obtained by filtering the output of this filter so that the Fourier analyzer would not see the pronounced stairstep pattern from the output DAC of the Datel 256 system. This was done by using a simple one-pole low pass filter using an operational amplifier. The filter was designed to have a cutoff frequency of 200 Hz, well above the active range of the differentiator, but enough to round off the output function somewhat.

The surprizing part of this combination of the digital and analog filters is that the phase was made even more positive and the phase roll-off was far more gradual. This shows that combinations of digital and analog filters may give results which are not what intuition would predict.

SECTION 12

RESULTS OF THE INFINITE IMPULSE RESPONSE FILTER COEFFICIENT SYNTESIS PROGRAM

Two types of filters have been run both on the Burroughs 6700 system and the pdp 11/45 system and two others have been run only on the Burroughs 6700 system. The types of filters which have been run on both systems are a low pass sine Butterworth filter and a bandpass tangent Chebychev type II filter.

Two others have been run on the Burroughs 6700 system only so far. After we slow down the execution times so that we can successfully verify the operation of the first two filters we will begin to verify these other two filters in real-time using the time/data Fourier Analyzer system.

We have used the following technique to verify the filter. We have set a sample rate of typically one Hertz of the digital filter and a sample rate either equal to this or one-tenth of this for the sample frequency of the Fourier Analyzer. Since there is no such thing as a digital anti-aliasing filter, we have used an analog filter for anti-aliasing. We also verified that without this analog filter that aliasing was a serious problem as expected by watching the inputs and outputs on the Tektronix storage oscilloscope. The difference frequencies between that of the sine wave signal generator and the Nyquist folding frequency for the given digital filter sample frequency were very obvious on the scope. A matched set of filters was used on both the input and the output so that the phase effects of the two filters would cancel. Otherwise, we would introduce lead by putting the filter on the input only. The filters have been previously tested using the random noise generator and the time/data system to verify that they are indeed matched in phase response over the bandwidth of interest. This has been done periodically by Mr. Bill Simmons. An analog anti-aliasing filter cutoff frequency has been chosen equal to the Nyquist folding frequency for sample rates of the time/data system equal to that of the digital filter execution program. The cutoff frequency is scaled down by an order of magnitude for correspondingly lower time/data sample rates. This is because the analog anti-aliasing effects if this were not done. Unfortunately we have not found any references, theoretical or empirical, regarding the effects of aliasing on the phase portion of the transfer function.

Returning to the matter of examining the results of the MAC/FIL program, let us first look at three examples of the lowpass sine Butterworth example. This program was run by Airman Lear for three cutoff frequencies, 1 Hz, 5 Hz, and 10 Hz, all for the same sample rate of 100 Hz. The cutoff frequencies, are, of course scaled by the sample frequency if it is desirable, and in our case, necessary to select another sample rate.

The tables of coefficients for the 1 Hz case are shown in figure 12-1. It can be seen here that two sets of coefficients are available here. Only the combined form coefficients have been tested so far on an execution program. The cascade implementation would be more efficient, but it is not nearly as general, and would hence require far more changes in the execution program for implementing different types of filters. The combined form has been speeded up where possible in the execution program by taking advantages of the symmetric properties of the filter to cut down the number of multiplications necessary in each loop where there exist symmetrical properties.

All of the tables show a Nyquist folding frequency of 50 Hz since the sample rate is constant and equal to 100 Hz.

Figure 12-2 shows the amplitude portion of the transfer function. There are certainly no surprises here. Just the transfer function for a lowpass filter. But figure 12-3 contains a surprise in the phase portion of the transfer function. Rather than finding just a simple phase rolloff as expected, we see that the phase first increases, characteristic of an analog bandpass filter, and then rolls off. This is a significant phase lead. This would be very useful if we could reproduce this result in real-time for isopad control. However, this has so far eluded us.

We now have the MAC/FIL program working properly. This program was purchased from Agbabian Associates by FJSRL several years ago. Problems encountered in using the program show that it was not operational at the time we began trying to use it several months ago. We would like to thank Airman Lear and Capt. Perry Cole for their efforts in getting this program working properly. The first few runs have been devoted to verifying the sample sets of data given in the MAC/FIL manual from Agbabian Associates. Complete agreement down to the least significant digits in the coefficients has been attained, thanks to the 96 bit double precision mode of the Burroughs 6700.

```
MAC/FIL                                                                                    PAGE NUMBER   1

MAC/FIL CASE                         1.
TYPE                                 LOWPASS.
IMPLEMENTATION                       SINE.
VARIETY                              BUTTERWORTH.
NUMBER OF RECURSIVE WEIGHTS          5.
SAMPLING INTERVAL                       0.0100 SECONDS.
NYQUIST FOLDING FREQUENCY              50.0000 HZ.
CUTOFF                                  1.0000 HZ.
NUMBER OF ORDINARY WEIGHTS           1.
STABILITY CHECK                      .  .883841E-05. RESULT IS MARG.
NUMBER OF BITS =                     10.23.

COMBINED FILTER

 I          H(I)        K          G(K)

 1.  .47057435E+01      0  .88388238E-06
 2  -.92074773E+01      1 0.
 3   .90407053E+01      2 0.
 4  -.44599902E+01      3 0.
 5   .81402729E+00      4 0.
                        5 0.

CASCADE FILTER

 I       H1(I)          H2(I)          G1(I)          G2(I)          G3(I)      STABILITY   DISTANCE     BITS

 1    1.95203151    -0.96100210     0.30050601     0.00000000     0.00000000     GOOD     .3875E-02     .55
 2    1.89050139    -0.90338230     0.30050501     0.00000000     0.00000000     GOOD     .3755E-02     .[illegible]
 3    0.93012077     0.00000000     0.30050501     0.00000000     0.00000000     GOOD     .6098E-01     .[illegible]
```

Figure 12-1 Coefficients for the 1 Hz Lowpass Butterworth.

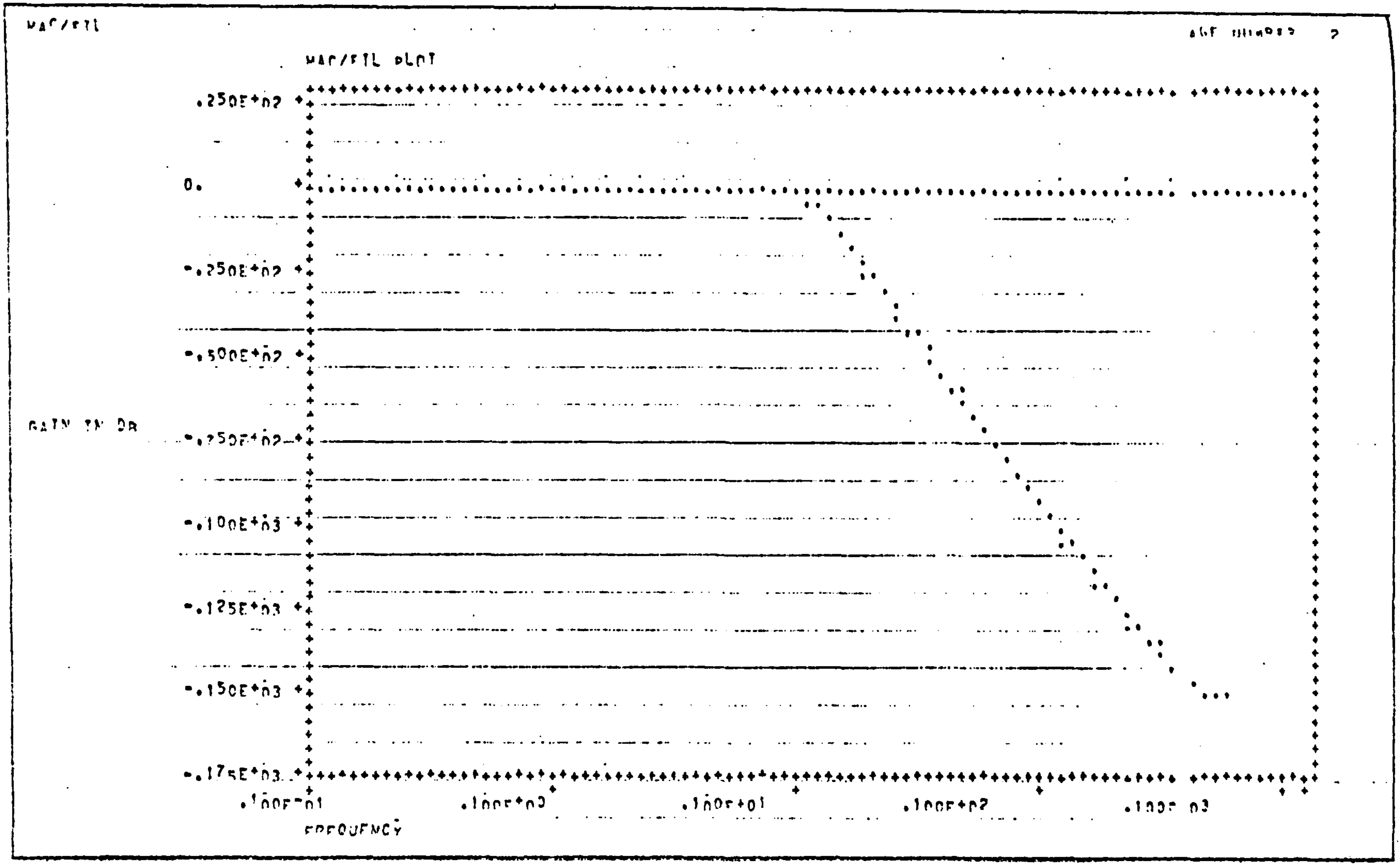


Figure 12-2 Transfer Function Magnitude for the 1 Hz Lowpass Butterworth.

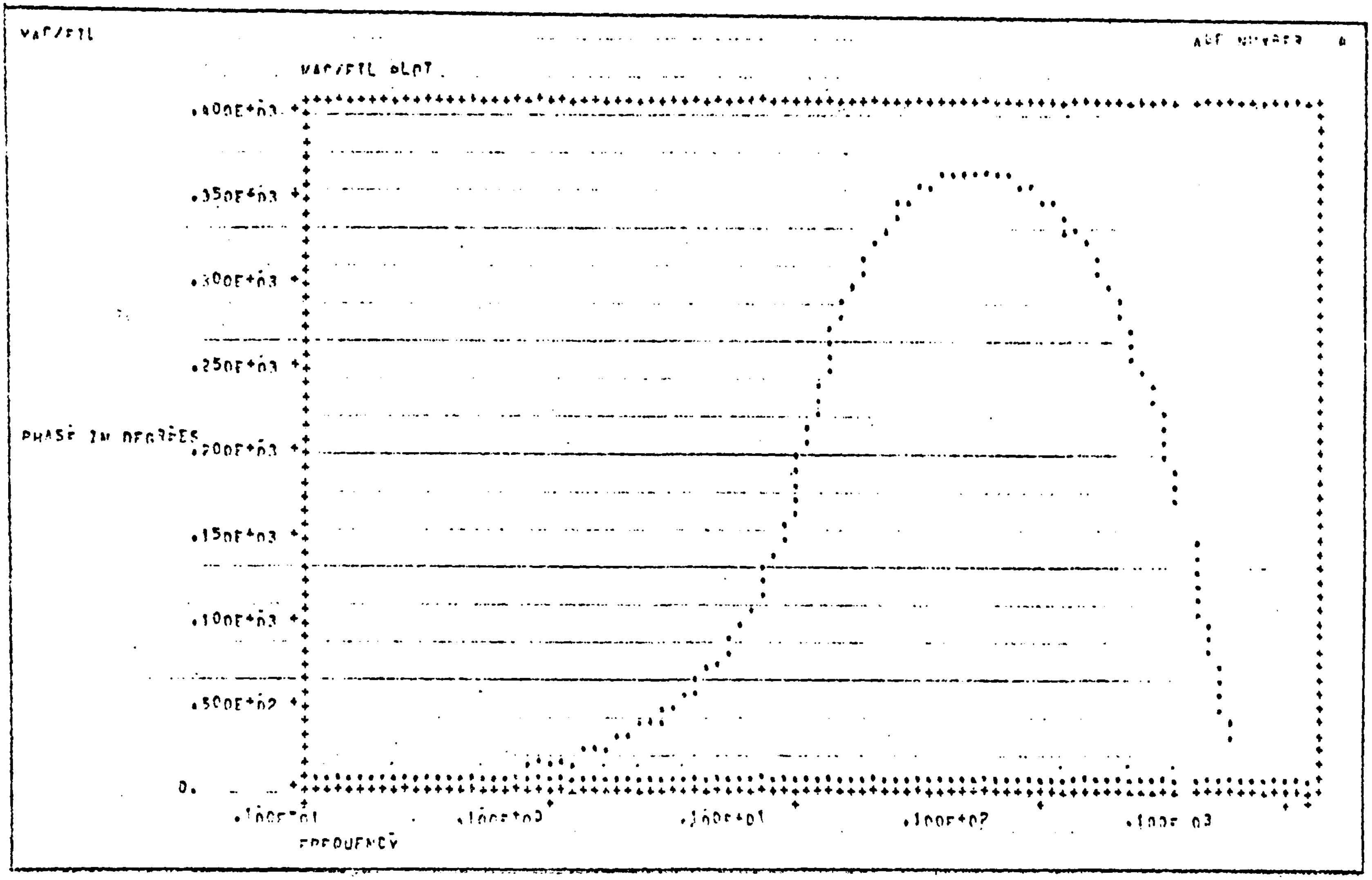


Figure 12-3 Transfer Function Phase for the 1 Hz Lowpass Butterworth.

These results could not be obtained on the pdp 11/45 system since the double precision arithmetic mode uses only 64 bits and roundoff errors prevented the calculation of any coefficients whatsoever.

The lowpass Butterworth filter coefficients for the cutoff of 5 Hz are shown in figure 12-4. The amlitude and phase portions of the transfer functions are shown in figures 12-5 and 12-6 respectively. The coefficients for the 10 Hz case are shown in figure 12-7 and the corresponding transfer function plots for this filter are shown in figures 12-8 and 12-9.

It can be seen in these three sets of plots that the lowpass rolloff is just what one might expect from an analog filter. However, the phase plots are distinctly different. Rather than getting just phase lag which goes down from zero to 90 degrees, we get a phase lead followed by lag which together forms a peak of phase. This would be decidedly more useful than the analog filter for isopad control if we can reproduce this phase transfer function in real-time.

One of the other features of these plots which seems strange to those accustomed to working with analog filters is that the shape of the curves change with the cutoff frequency. With the analog filters one would expect the curves to shift with frequency, but not change in shape.

The next set of data shows the coefficients and the transfer function curves generated by the Burroughs 6700 for the case of a tangent implementation of a bandpass Chebychev type II filter. This filter was chosen because it was suspected that the band of this filter where the amplitude portion of the transfer function is increasing would yield phase lead.

The data for this filter is shown in figures 12-10, 12-11, and 12-12. The region where this filter displays lead covers nearly two decades of frequency and is very smooth. This gives us hope that it may be useful for isopad control, since it should be stable and yet give a large area of phase lead. Thus both the lowpass Butterworth filter and the bandpass Chebychev filter give significant bands of phase lead yet exhibit good stability. If these results can be reproduced in real-time this will be far better than the performance obtained by any similar analog filters.

Figures 12-13, 12-14, and 12-15 show the coefficients and transfer function plots for a notch filter. This is of interest since one of the filters Emil

MAC/FIL

ASE NUMBER 1

MAC/FIL CASE .... 1.
TYPE .... LOWPASS.
IMPLEMENTATION .... SINE.
VARIETY .... BUTTERWORTH.
NUMBER OF RECURSIVE WEIGHTS . 5.
SAMPLING INTERVAL .... 0.0100 SECONDS.
NYQUIST FOLDING FREQUENCY ... 50.0000 HZ.
CUTOFF .... 5.0000 HZ.
NUMBER OF ORDINARY WEIGHTS .. 1.
STABILITY CHECK .... .1803564E-02. RESULT IS GOOD.
NUMBER OF BITS F .... 8.69.

COMBINED FILTER

I	U(I)	K	D(K)
1	.3902290055E+01	0	.1805562AE-02
2	-.6495816255E+01	1	0.
3	.5280585516E+01	2	0.
4	-.2179226316E+01	3	0.
5	.3627648435E+00	4	0.
		5	0.

CASCADE FILTER

I	H1(I)	H2(I)	G1(I)	G2(I)	G3(I)	STABILITY	DISTANCE	ITS
1	1.73359565	-0.82965887	0.12174722	0.00000000	0.00000000	GOOD	.888E-01	.67
2	1.52433549	-0.68230384	0.12176922	0.00000000	0.00000000	GOOD	.780E-01	.47
3	0.733[illegible]987	0.00000000	0.12174922	0.00000000	0.00000000	GOOD	.265E+00	.72

Figure 12-4 Coefficients for the 5 Hz Lowpass Butterworth Filter.

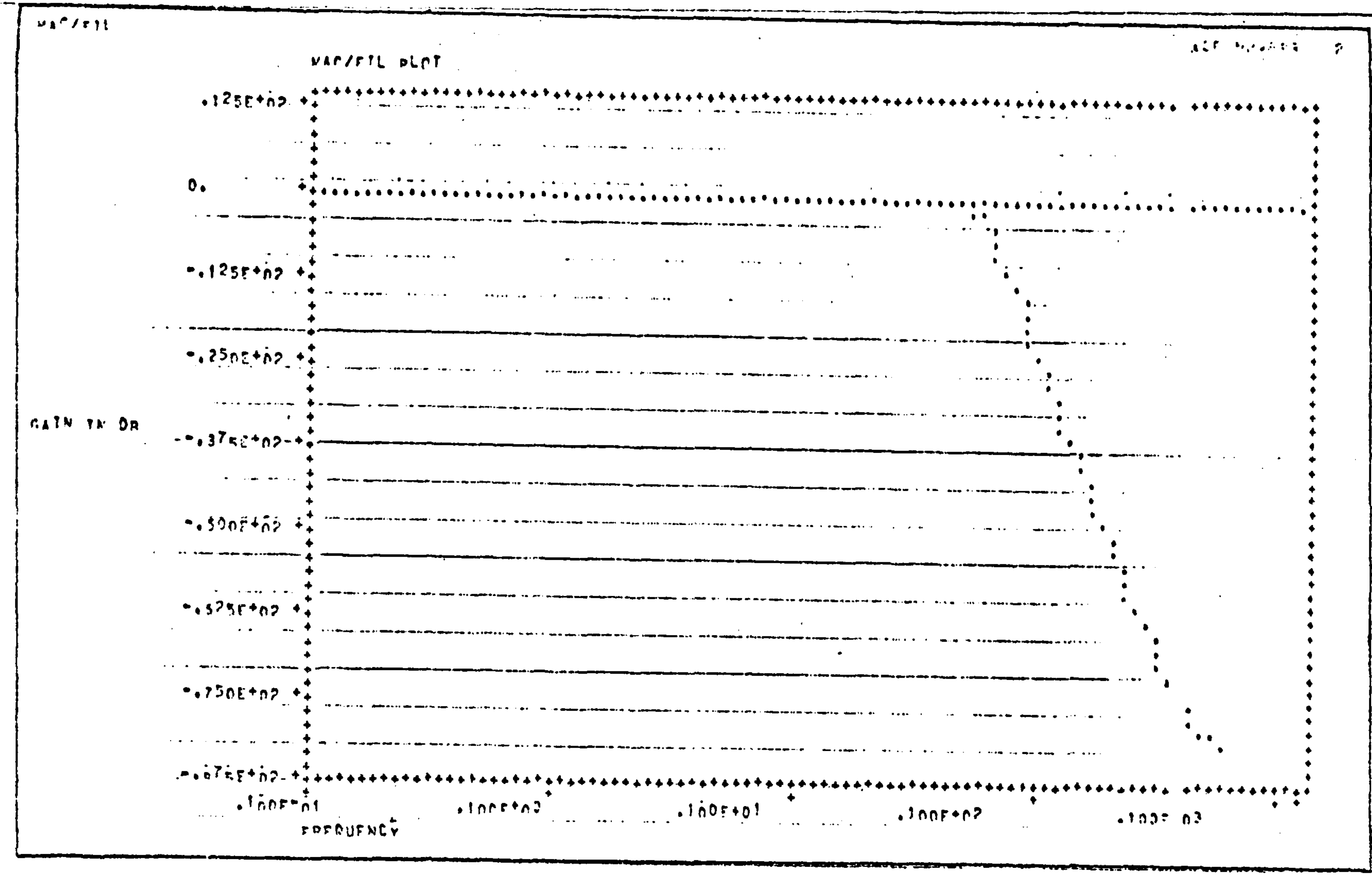


Figure 12-5 Transfer Function Magnitude for the 5 Hz Lowpass Butterworth.

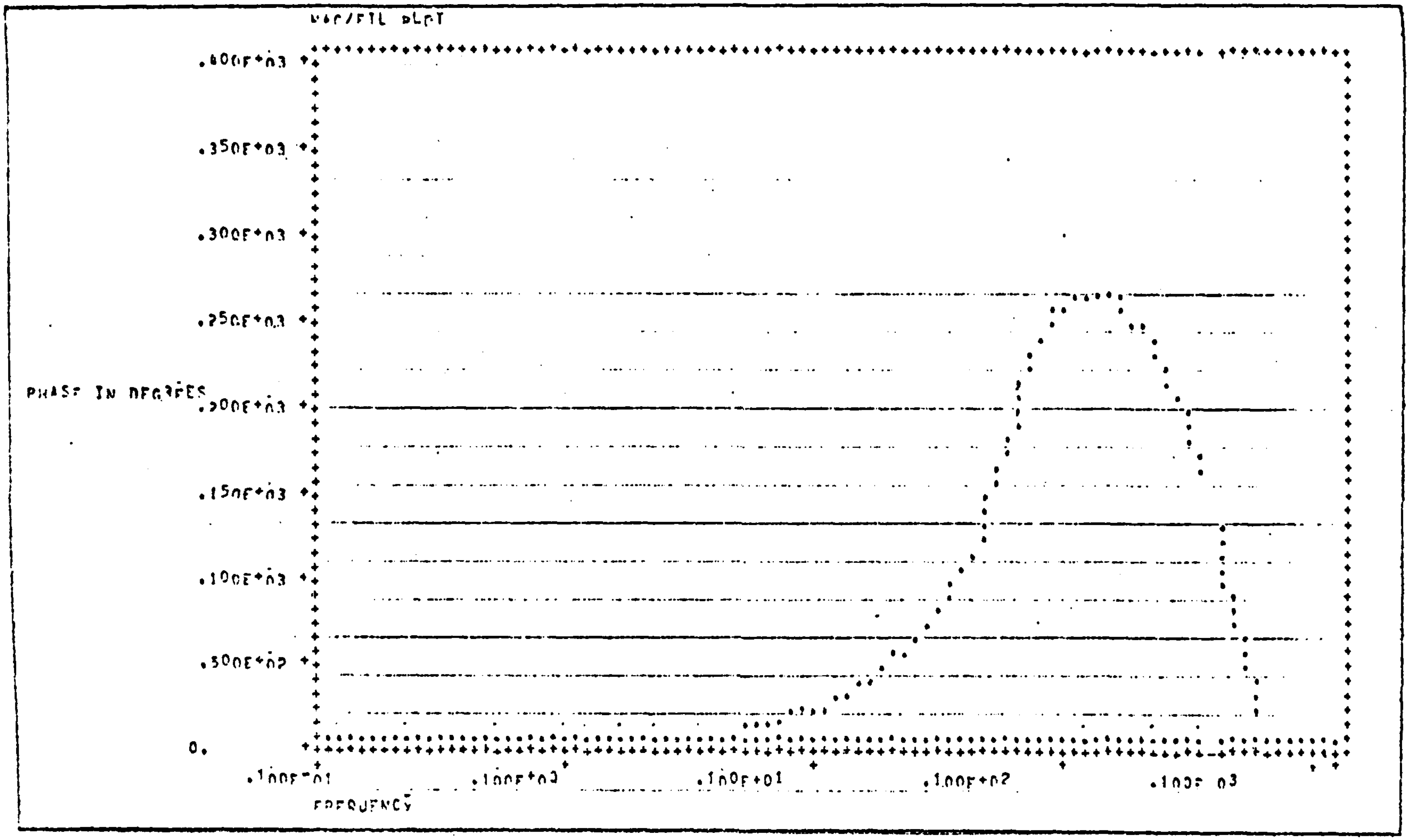


Figure 12-6 Transfer Function Phase for the 5 Hz Lowpass Butterworth.

```
MAC/FIL                                                                                    PAGE NUMBER   1

MAC/FIL CASE                          1.
TYPE                                  LOWPASS.
IMPLEMENTATION                        SINE.
VARIETY                               BUTTERWORTH.
NUMBER OF RECURSIVE WEIGHTS           5.
SAMPLING INTERVAL                        0.0100 SECONDS.
NYQUIST FOLDING FREQUENCY               50.0000 HZ.
CUTOFF                                  10.0000 HZ.
NUMBER OF ORDINARY WEIGHTS            1.
STABILITY CHECK                       .32953157E-01, RESULT IS GOOD.
NUMBER OF BITS =                      4.62.

COMBINED FILTER

 I          H(I)           K         G(K)

 1   .30372265E+01         0  .32963093E-01
 2  -.39350605E+01         1 0.
 3   .26617332E+01         2 0.
 4  -.93049618E+00         3 0.
 5   .13353872E+00         4 0.
                           5 0.

CASCADE FILTER

 I        H1(I)          H2(I)          G1(I)          G2(I)          G3(I)      STABILITY   DISTANCE    ITS

 1      1.35823721    -0.67116760     0.32063381     0.00000000     0.00000000      GOOD     .313E+00    .54
 2      1.13438613    -0.36595279     0.32063381     0.00000000     0.00000000      GOOD     .231E+00    .70
 3      0.58411322     0.00000000     0.32063381     0.00000000     0.00000000      GOOD     .455E+00    .15
```

Figure 12-7 Coefficients for the 10 Hz Lowpass Butterworth.

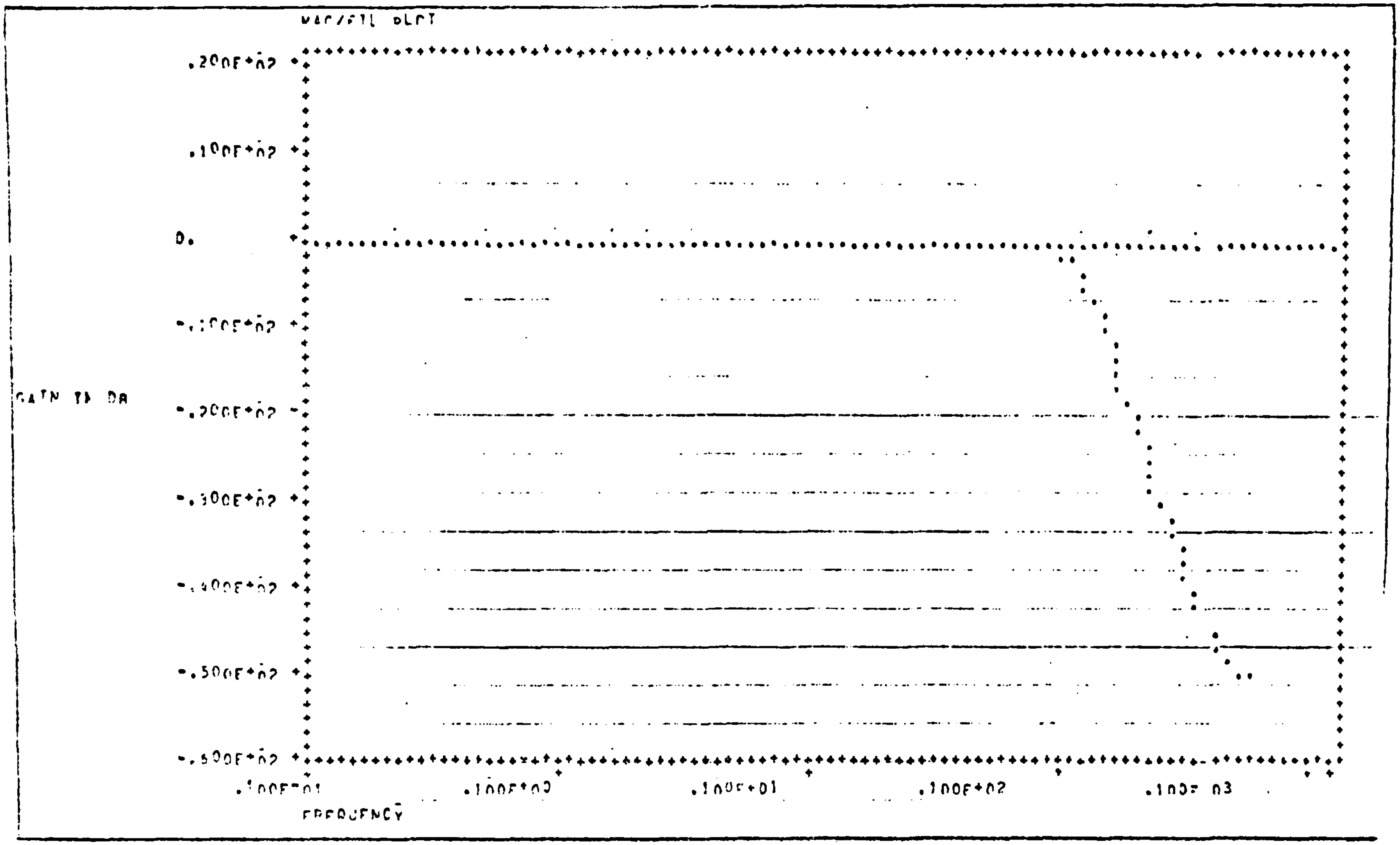


Figure 12-8 Transfer Function Magnitude for the 10 Hz Lowpass Butterworth.

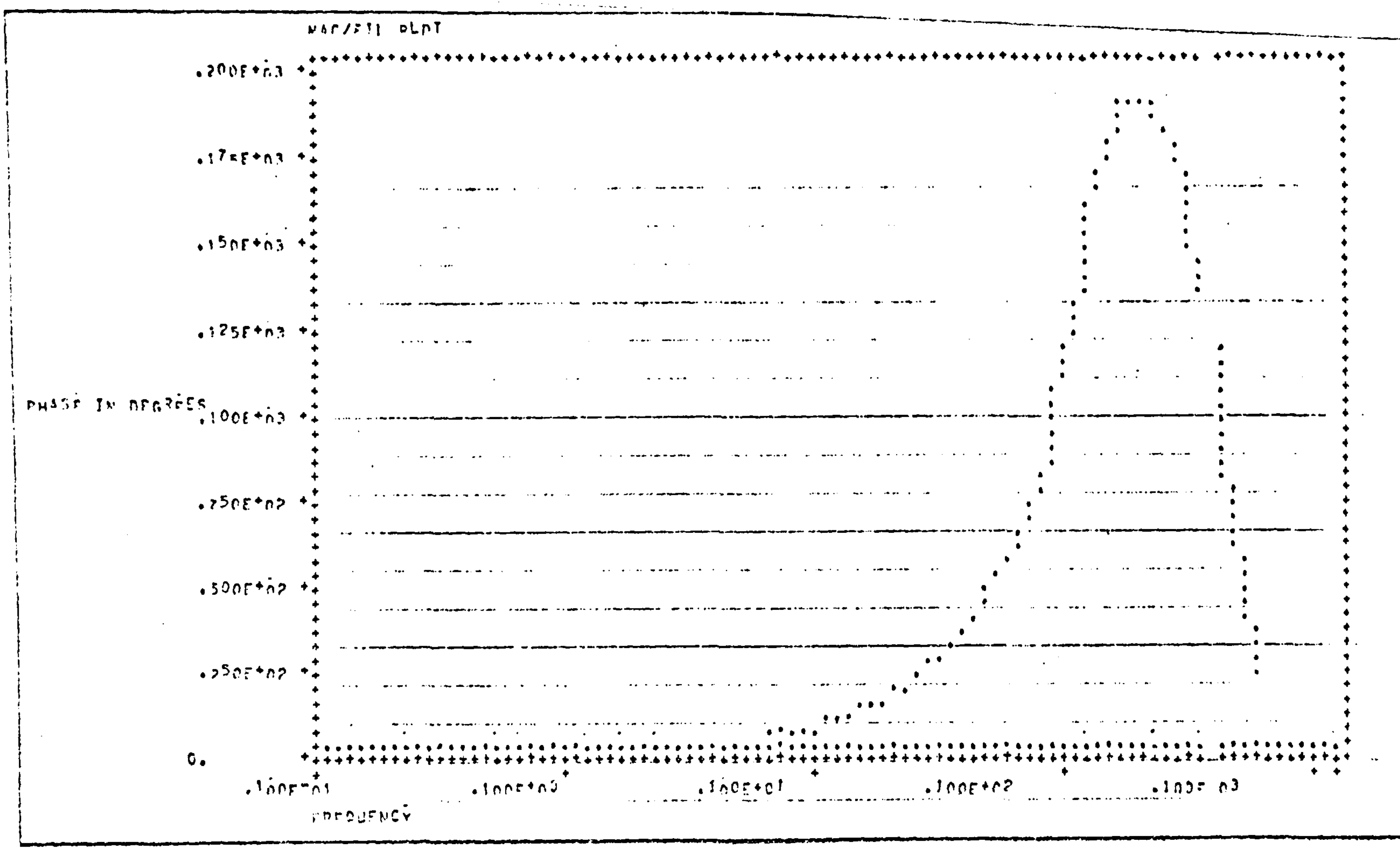


Figure 12-9 Transfer Function Phase for the 10 Hz Lowpass Butterworth.

MAC/FIL

MAC/FIL CASE 1.
TYPE BANDPASS,
IMPLEMENTATION TANDEM,
VARIETY CHEBYSHEV TYPE II
NUMBER OF RECURSIVE WEIGHTS 8.

PAGE NUMBER 1

I	W(I)	K	G(K)
1	.74013017E+01	0	.14973961E-02
2	-.24070299E+02	1	-.87154356E-02
3	.44022981E+02	2	.23574271E-01
4	-.52430645E+02	3	-.39814594E-01
5	.39425217E+02	4	.46916724E-01
6	-.17724005E+02	5	-.39814594E-01
7	.51086529E+01	6	.23574271E-01
8	-.66460113E+00	7	-.87154356E-02
		8	.14973961E-02

CASCADE FILTER

I	H1(I)	H2(I)	G1(I)	G2(I)	G3(I)	STABILITY	DISTANCE	BITS
1	1.83026793	-0.90222021	0.13671351	-0.39337486	0.19671351	GOOD	.721E-01	4.08
2	1.75333825	-0.70520529	0.13671351	-0.39315552	0.19671351	GOOD	.417E-01	4.00
3	1.66231325	-0.67665813	0.13671351	-0.05582307	0.19671351	GOOD	.143E-01	4.09
4	1.95520224	-0.96434631	0.13671351	-0.20555067	0.19671351	GOOD	.914E-02	4.18

Figure 12-10 Coefficients for the 1.5 Hz to 4.5 Hz Bandpass Chebyshev.

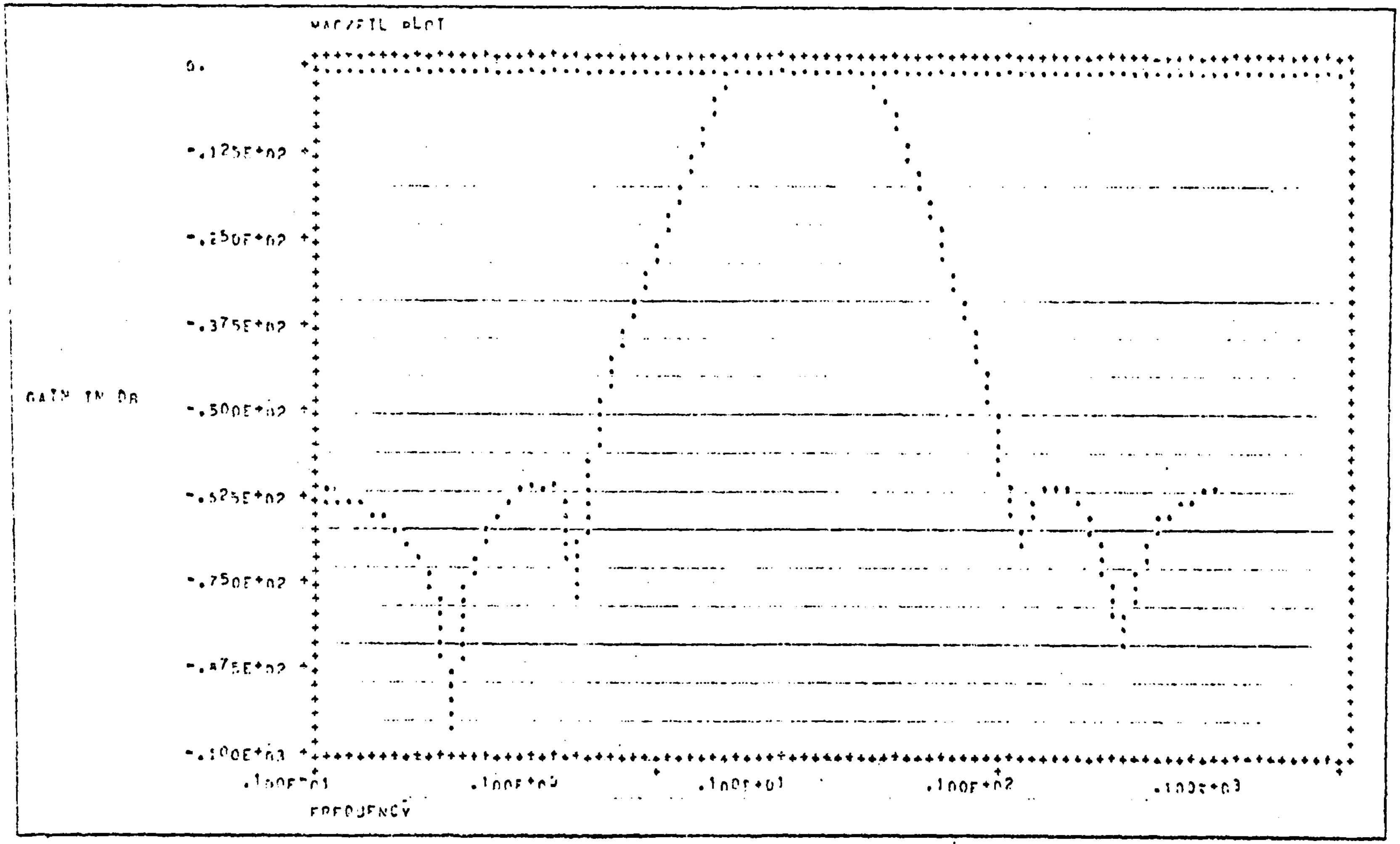


Figure 12-11 Transfer Function Magnitude for the Bandpass Chebyshev.

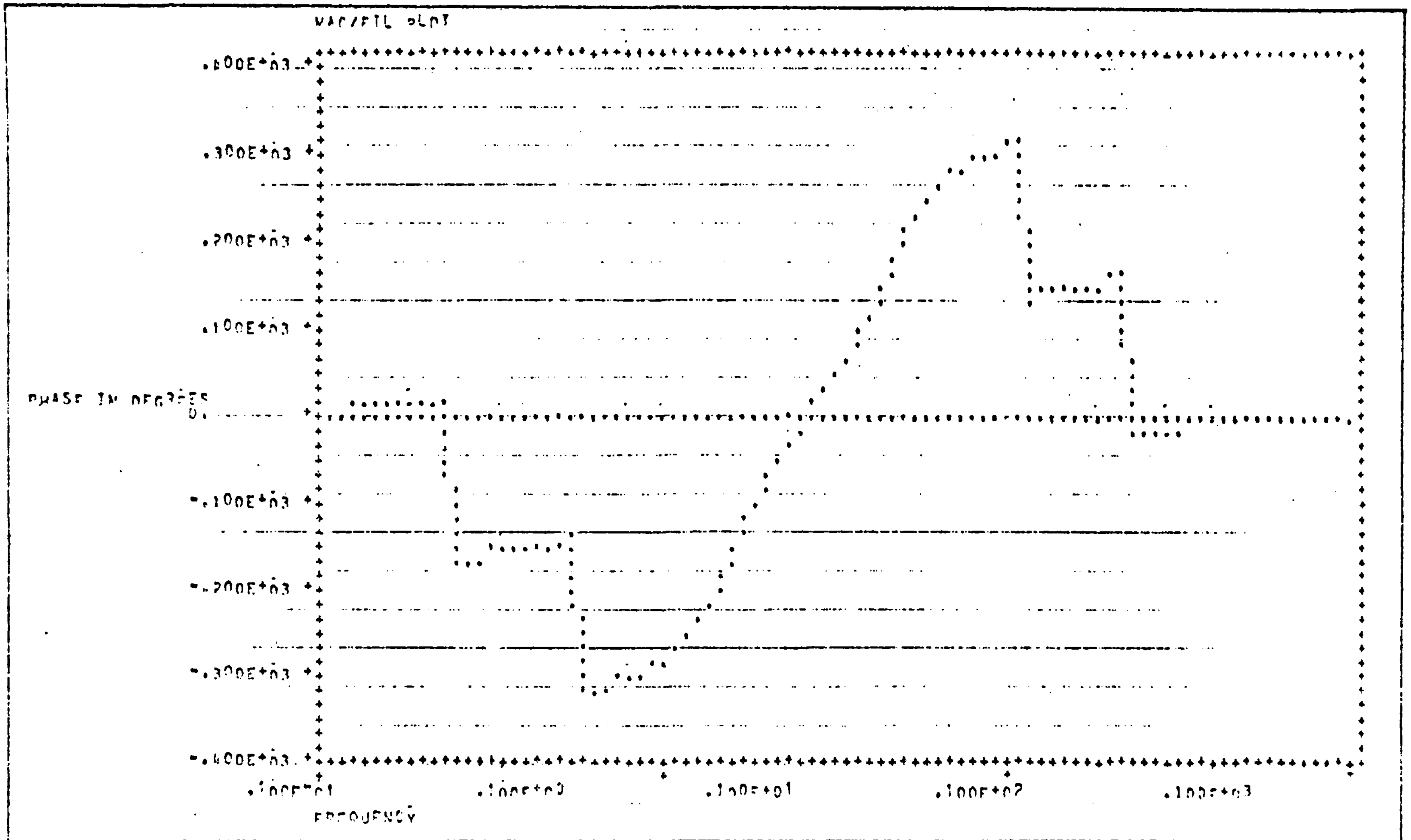


Figure 12-12 Transfer Function Phase for the Bandpass Chebyshev.

```
MAL/FIL CASE                        1.
TYPE                                NOTCH.
IMPLEMENTATION                      SINE.
VARIETY                             BUTTERWORTH.
NUMBER OF RECURSIVE WEIGHTS         6.
BAND REJECT                             1.5000 TO      4.5000 HZ.
SAMPLING INTERVAL                       0.0100 SECONDS.
NYQUIST FOLDING FREQUENCY              50.0000 HZ.
NUMBER OF ORDINARY WEIGHTS          7.
NUMBER OF BITS =                    0.38.
```

COMBINED FILTER

I	H(I)	K	G(K)
1	.55200890E+01	0	.94000020E+00
2	-.12623642E+02	1	-.55157468E+01
3	.16041784E+02	2	.13698129E+02
4	-.11358537E+02	3	-.18054760E+02
5	.43586276E+01	4	.13605129E+02
6	-.70130267E+00	5	-.55157468E+01
		6	.94000402E+00

CASCADE FILTER

I	H1(I)	H2(I)	G1(I)	G2(I)	G3(I)	STABILITY	DISTANCE	BITS
1	1.85554103	-0.87638925	0.97960007	-1.91595634	0.97960007	GOOD	.208E-01	3.56
2	1.83149636	-0.83370278	0.97960007	-1.91595634	0.97960007	GOOD	.552E-01	3.51
3	1.86305164	-0.937[illegible]94	0.97960007	-1.91595634	0.97960007	GOOD	.627E-01	3.57

Figure 12-13 Coefficients for the Butterworth Notch Filter.

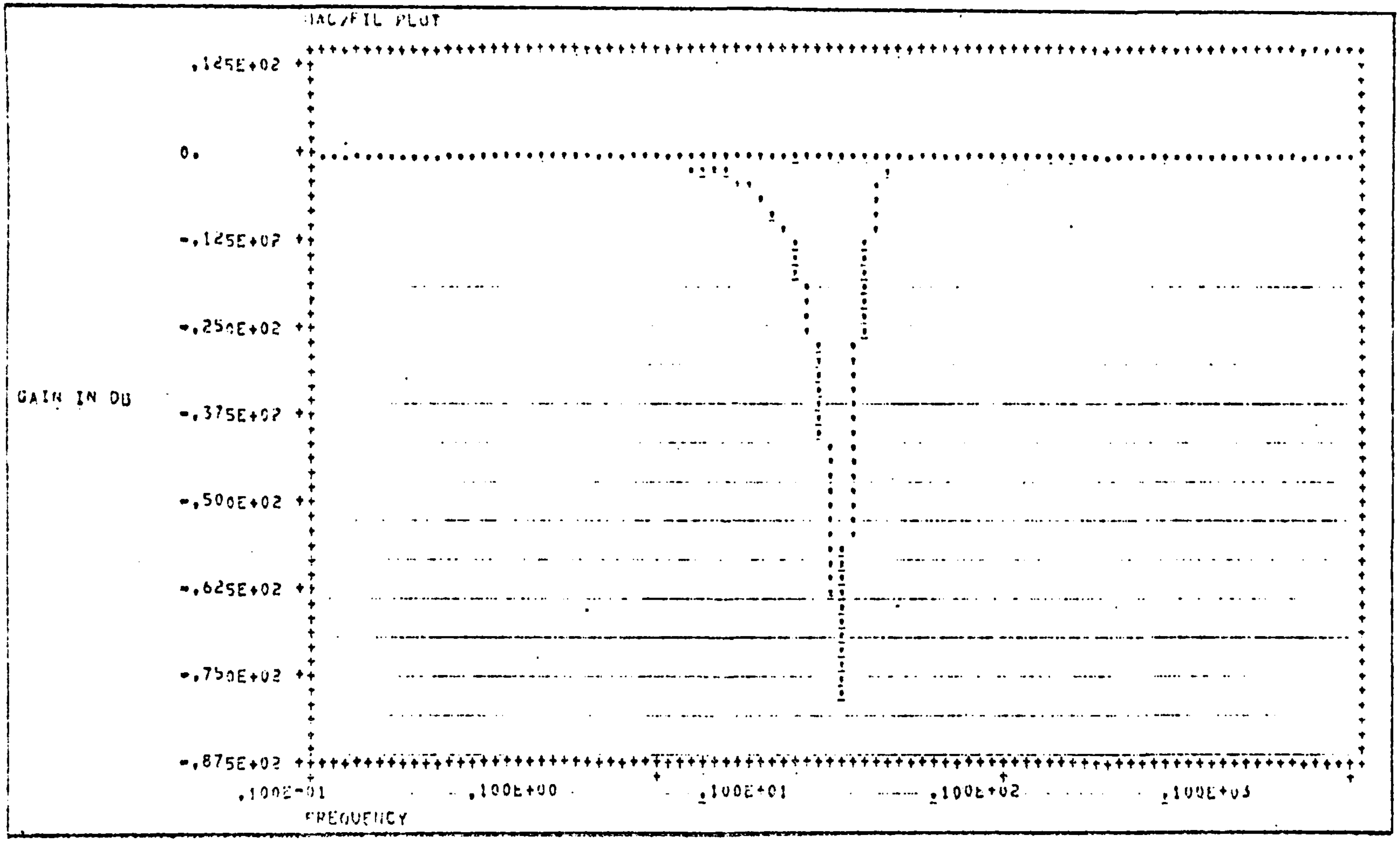


Figure 12-14 Transfer Function Magnitude for the Butterworth Notch Filter.

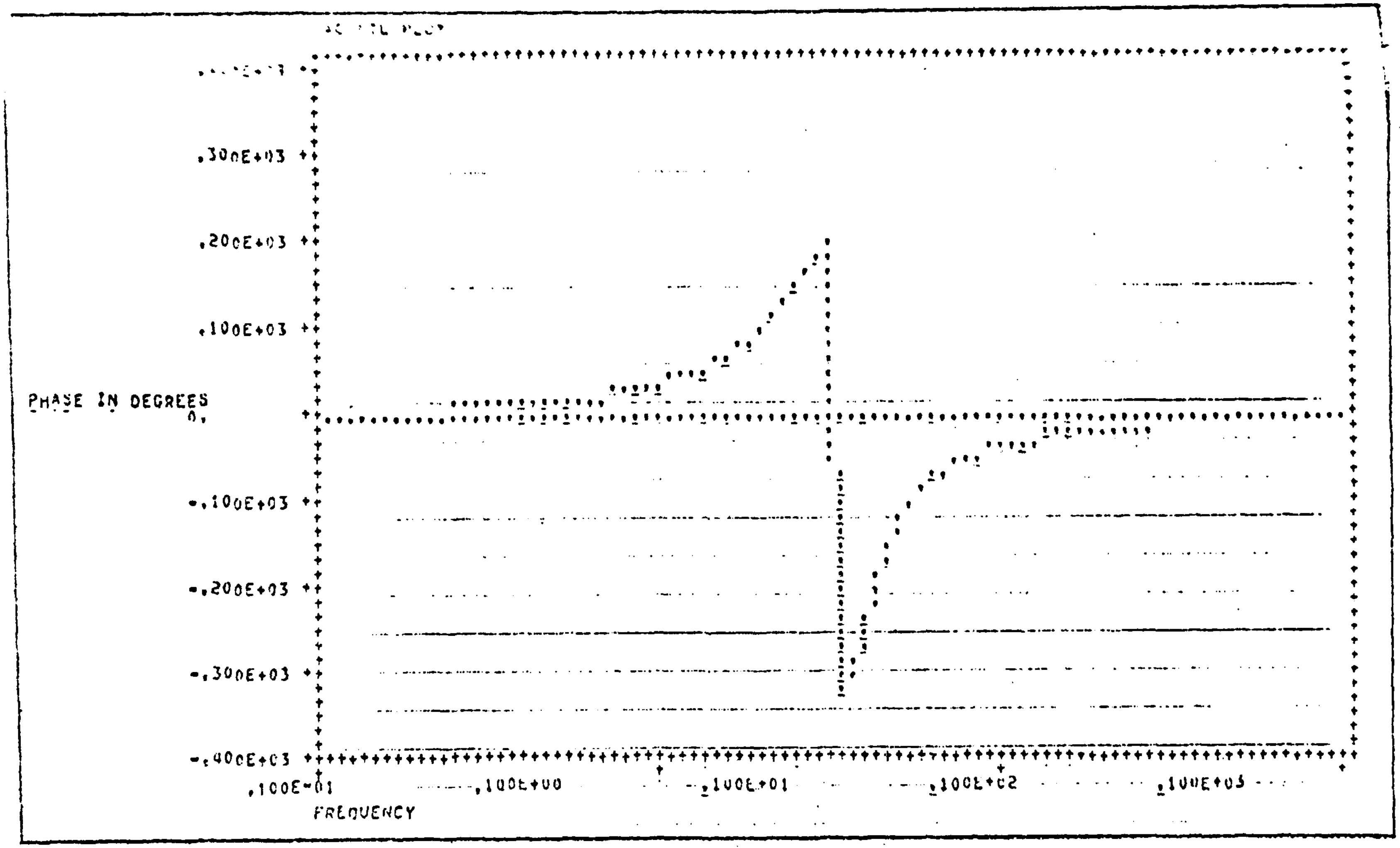


Figure 12-15 Transfer Function Phase for the Butterworth Notch Filter.

Brodcrson found useful for isopad control was a similar band reject or notch filter. This is because the phase lead comes at a point where the stability can easily be controlled. The digital implementation of the filter shows lead too, but it also shows a phase inversion relative to the analog case. The region of phase lead in this filter is not nearly so spread out and smooth as in the previous two cases.